# Investigating the predictability of a Chaotic time-series data using Reservoir computing, Deep-Learning and Machine- Learning on the short-, medium- and long-term pricing of Bitcoin and Ethereum.

**MOLLY KENNY**

*D19123736*

A dissertation submitted in partial fulfilment of the requirements of
Technological University Dublin for the degree of
M.Sc. in Computer Science (Data Science)

**2020**

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Data Science), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the test of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Technological University Dublin and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

**Signed:** Molly Kenny

**Date:** **30 September 2020**

## ABSTRACT

This study will investigate the predictability of a Chaotic time-series data using Reservoir computing (Echo State Network), Deep-Learning(LSTM) and Machine-Learning(Linear, Bayesian, ElasticNetCV , Random Forest, XGBoost Regression and a machine learning Neural Network) on the short (1-day out prediction), medium (5-day out prediction) and long-term (30-day out prediction) pricing of Bitcoin and Ethereum

Using a range of machine learning tools, to perform feature selection by permutation importance to select technical indicators on the individual cryptocurrencies, to ensure the datasets are the best for predictions per cryptocurrency while reducing noise within the models.

The predictability of these two chaotic time-series is then compared to evaluate the models to find the best fit model. The models are fine-tuned, with hyperparameters, design of the network within the LSTM and the reservoir size within the Echo State Network being adjusted to improve accuracy and speed.

This research highlights the effect of the trends within the cryptocurrency and its effect on predictive models, these models will then be optimized with hyperparameter tuning, and be evaluated to compare the models across the two currencies.

It is found that the datasets for each cryptocurrency are different, due to the different permutation importance, which does not affect the overall predictability of the models with the short and medium-term predictions having the same models being the top performers.

This research confirms that the chaotic data although can have positive results for short- and medium-term prediction, for long-term prediction, technical analysis based-prediction is not sufficient.

**Keywords:** *Chaotic time-series, Cryptocurrency, Echo State Network, Price forecasting, reservoir computing, Neural Network*

# ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my supervisor Brendan Tierney, for his guidance, understanding and humour through the course of this dissertation.

I would also like to thank all of my TU Dublin lecturers for assisting me in gaining the skills and knowledge to complete this dissertation and to bring into my future works, not only did I develop the technical skills through this Masters but the critical thinking and innovation skills that will remain with me for life.

My deepest thanks go to my parents and siblings, I don't think any of us expected to be in the same house for several months of the year and their support through the final months of my thesis and dissertation process was incredible.

Finally, I would like to thank my fantastic friends who have supported me through this entire year.

# TABLE OF CONTENTS

## Table of Figures

# TABLE OF TABLES

# 1. INTRODUCTION

This chapter presents the background of the research topic and identifies the research problem while outlining the importance of this area.
This is followed by the research question including the research hypothesis, the aims and objective and the research methodologies.
The outline of the scope and limitations of the study will then be identified, and the chapter will end by outlining the rest of the document.

## 1.1 Background on the data

Cryptocurrencies have increased in popularity since the publication of Bitcoin in 2009, and its start of active trading 2013, a particularly noteworthy time in the cryptocurrency markets which highlight the "Cryptocurrency boom in 2017". Although there has been a large increase in their use, the research in the area and the trading of these currencies, they remain incredibly volatile and therefore difficult to predict. This is due to the fact that they are available to trade 24/7, decentralized, and the mining activity is unmonitored.

Bitcoin and Ether are currently the top 2 ranking cryptocurrencies on the market, with a combined market cap of $192.05 billion U.S dollars at the end of June 2020.
Both are block-chain, decentralized cryptocurrencies, the distinct difference between the cryptocurrencies is the mining approach, the mission behind their founding and the block-time, where ether transactions are confirmed within seconds, it can take several minutes for Bitcoin. Ether was established to be a complement Bitcoin, yet has nonetheless become its main competitor for market cap.

Ether is the native language of Ethereum, a blockchain technology platform, "the world's programmable blockchain", it was released in 2015, it utilizes block-chain technology not just as a decentralized payment network but can be used to power decentralized financial contracts and applications. Decentralized Application (dapp) platform.

Ether can be traded in the same way as Bitcoin, a tradable commodity but also as a payment to use the Ethereum network to run applications. In this way, transactions on the Ethereum network can contain executable code, while data on the Bitcoin network are only for keeping notes, due to the languages used with Ethereum being etash while Bitcoin uses SHA-256, which is infinitely more difficult to imagine coding in than ethash.

Within (Rauchs et al., 2018) study there are now over 139million user accounts with service providers, with at least 35 million identity-verified users, with growth of 4X in 2017 and a doubling in 2018. Although there are only about 38% of users who are considered active, with multi-coin activity rapidly expanding.

It can be seen within the Chainalysis, “What is going on with the Bitcoin Market” [1] that within early March 2020 there was a “an unprecedented inflow of cryptocurrency to exchanges in response to the COVID-19 pandemic”. The report highlights that from Jan 1st- March 9th, 2020, an average of 52,000 bitcoin per day were received by exchanges, on March 13th, 2020 that peaked at 312,000 bitcoins. There was also 9x the daily average bitcoin sent to exchanges to be sold from March 12th to March 13th, this sell pressure led to a 37% fall in the price of Bitcoin.

The Bitcoin and cryptocurrency market are now seen to be a large market trading commodity, this reduces the volatility in the market, by professional traders and investors (“whales”) taking a larger portion of the coins. Therefore, the market is more controlled by professionals than it was initially by smaller investors. Although there was a large increase in the small transfers, which is between 0.1-10 bitcoin, doubled between March 9th - March 18th, although “transfers between 10 and 1,000 bitcoin were responsible for 70% of the bitcoin through exchanges” (Gradwell, 2020)

Global Exchange transactions for Bitcoin within Jan ’19 – Jun ’19, provides an insight into where Bitcoin was entering the exchanges from, as there is no other way to cash out your bitcoin for cash than go through an exchange, this is the diagram shows where the money came from to go into the exchange. With the majority of the money entering

[1] https://blog.chainalysis.com/reports/bitcoin-market-march-2020

being from other exchanges, it is noteworthy that there is such a large portion in "uncategorized", highlighting how cryptocurrencies are still truly a secret currency.

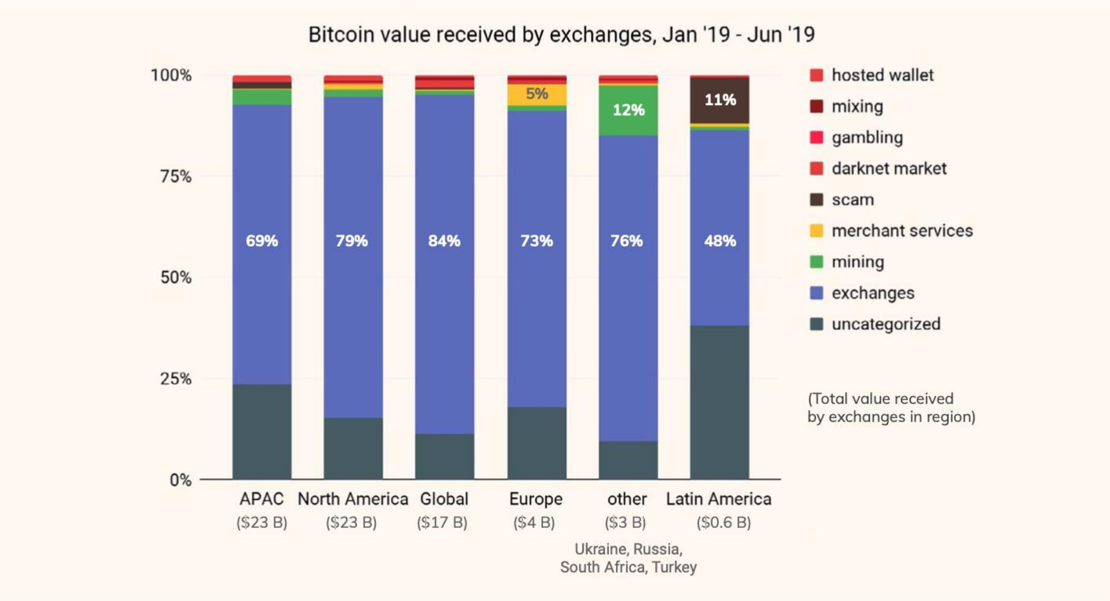


**Figure 1.1 Global Exchange transactions for Bitcoin within Jan '19 – Jun '19.**

## 1.2 Background on the models

Predicting the stock markets is an extremely lucrative area for both investment institutions, governments and the shareholders of companies, being able to predict the future price of anything can encourage you to sell, buy, short or long a stock.

Within the last 20 years, with the revolution of online trading platforms, where you need no qualification or broker to buy and sell stocks, foreign currencies, commodities and cryptocurrencies, this has led to the further interest in algorithms and predictive models for the general public, as well as increased the demand within financial institutions for algorithmic trading by technical staff rather than by economists.

Both technical and fundamental trading are vital for as models are never going to be able to gather all information and see its weighting with just technical analysis, particularly of note is the growth of sentiment analysis models, which are attempting to incorporate public opinion which was a part of the fundamental analysis by reading the general public sentiment.

Neural networks have been proven to be a powerful tool in assisting with technical analysis, for both Cryptocurrencies and stock market prediction, (Sin & Wang, 2017), (Jang & Lee, 2018), (Guresen et al., 2011) using historical pricing to predict future pricing as directional and as values.

The blockchain also provides a powerful insight into what is happening with Bitcoin mining and therefore giving the basics of "supply/demand" information, as Bitcoin gets harder to mine, it will likely increase the value of the currency, (Jang & Lee, 2018), who conduct an empirical study on modelling and predicting the price of bitcoin, based on the Blockchain information, sentiment analysis using social and web search media is also a popular yet more unreliable way of predicting Bitcoin prices as seen in (Matta et al., 2015)

This study will review the predictive power of different neural networks on Bitcoin and Ether, technical indicators will be used to provide the network with as much information to the network without overfitting.
Due to the volatility of cryptocurrencies series neural network architectures will be investigated to find the most predictive model and compared with machine learning models, an Echo state, LSTM will be used to examine if the chaotic nature of cryptocurrencies can truly be predicted the short, medium and long term returns accurately.

## 1.3 Research Project/problem

This study will examine the use of machine learning and neural networks to predict Bitcoin and Ether prices, as cryptocurrencies are a relatively new financial product, reference texts will span into other financial products and state of the art predictive models in other areas, to examine the most effective way of predicting the returns price of this highly chaotic market.

Using exploratory analysis to provide the technical indicators and tuning the hyperparameters within each network, can the short-, medium- and long-term returns of Bitcoin and Ethereum be predicted with only using technical analysis?

**Null Hypothesis:** Bitcoin and Ethereum, cannot be predicted with Machine learning and Neural Network models, to a degree of accuracy for short-, medium-(closing price of the week) and long-term(closing price of the month) price.

**Alternate Hypothesis:** Bitcoin and Ethereum short-, medium- and long-term direction of pricing, can be predicted by only using technical analysis with Machine learning and Neural Network.

## 1.4 Research Objectives

Primarily, this research aims to determine the forecasting capabilities of an Echo State within a Neural Network and whether it will perform better than an LSTM model in forecasting the stock price direction, it will then be examined as to what level of accuracy the model can reach on exact price prediction for short, medium and long term returns of Bitcoin and Ether.

These models will be compared on Bitcoin and Ether historical data, as although there is a strong positive correlation between the cryptocurrencies currently, with the development of Ethereum 2.0, it is speculated that Ether will stop being as affected by Bitcoin price changes.

The study was consist of an initial set of 80 technical indicators, which will be examined and pruned, these features may differ for Bitcoin and Ether and will feed into each model.

The models will be evaluated to identify accuracy and MSE. This will provide the results of the null hypothesis.

To gain insight into the best performing model when used against the stock data the following tasks will be implemented:

- Study existing literature on crypto-market trends, crypto-market trading behaviour, market trends such as those of bull and bear, technical indicators, and machine learning models to gain an in-depth analysis of the research and tools used by academics and traders alike.
- Perform the feature selection and analysis of the overall data to clean and prepare it for modelling.
- Analysing the data to split into train and test samples.
- Calculate the future return price.
- Build the models to implement the data into the Echo State Neural network. and LSTM, machine learning models.
- Evaluate the model performance by utilizing the accuracy and MSE.

The aims and objectives of the research are:

- To critically analyse the literature regarding cryptocurrencies, predictive models and technical indicators used within the cryptocurrency and other financial markets.
- Statistically analyses the factors which affect the cryptocurrency markets and specifically Bitcoin and Ether.
- Evaluate the performance of ESN, LSTM and machine learning models for predicting the price return of the currencies in the short-, medium- and long-term.
- Provide empirical evidence to accept or reject the null hypothesis.

## 1.5 Research Methodologies

This research is a collective set of data that is measurable by mathematical expressions and quantitative methods. The mathematical models will consist of multiple machine learning algorithms which will test the best accuracy and evaluation when forecasting the crypto-price.

The data comprises of Bitcoin and Ether prices acquired from Yahoo Finance API, which is a platform which provides cryptocurrency markets data and insights.

The analysis will use a deductive reasoning approach on the secondary data from the cryptocurrencies Bitcoin and Ethereum to develop a Quantitative predictive model, to predict the return price, for the short (1-day), medium (5-day) and long (30-day) of both currencies.

## 1.6 Scope and Limitations

The full data period covers from 27/10/2015 up to the 25/08/2020, although the data is split into sections, to analyse if the data becomes less chaotic over time.

Technical and fundamental analysis are key to crypto-trading, to gain an understanding on the 2017 crypto-boom, There is a large amount of fundamental analysis a trader would investigate, such as the Chicago Stock Exchange market opening trading it, the SEC approval for funds to include cryptocurrencies within their portfolios, which all had an impact on the boom/bubble of 2017. Also within fundamental analysis key-dates, such as the May-Drop and SEC approval of Bitcoin will be examined in the graphs, to show their impact on the market and the ‘Bullish behaviour” like with Greyscale investments aggressive purchasing of Bitcoin post-may drop in June 2020, this data would have an impact on the target return, which would then be set by a professional trader, but will not have an effect on the technical models which will be produced in this study.

## 1.7 Document Outline

Within this study a discussion on the following will occur:

- Chapter two – "Literature Review", will conduct a review on past research within the area of Cryptocurrency, algorithms used within financial product prediction. This chapter will also outline the approached commonly used by traders and researched to accurately predict the cryptocurrency market.
- Chapter three – "Design and Methodology" will outline the method breakdown of the experiment. The design process followed is graphically outlined at the beginning of the chapter. Within this chapter, there is a special focus on the parameters used within the models and the hyperparameter optimization, which will refine each machine learning model.
- Chapter four – "Implementation and Results" provides a breakdown of each model implemented and the results of each stage of the experiment.
- Chapter five – "Evaluation" will be comprised of the result of each experiments along with the analysis on the relevance in relation to other works which have been examined in the literature review. This chapter encompasses the analytical aspect of the results and will confirm the disproof of the null-hypothesis.
- Chapter six – "Conclusion" will provide an overview of the entire study. Focusing on the experimental analysis place within the broader body of knowledge, examining the initial research question discussed in chapter one and provide insight into future work and recommendations.

# 2. LITERATURE REVIEW

Numerous studies have been conducted on modelling the time series of cryptocurrencies, within this literature review there will be a discussion on the cryptocurrency market and the place of cryptocurrencies as a financial product, reviewing current markets.
The literature review will then go into depth on the predictive models used currently within cryptocurrency prediction, focusing on neural network models, this chapter will end by looking at different trading strategies and how those strategies can be used in cryptocurrency prediction.

This chapter will then explore several types predictive models used within cryptocurrencies and other financial products, as predictive models are of such a large research interest in both academia and financial institutions, several studies will be explored to understand the effectiveness of Neural Networks and machine learning models on cryptocurrencies and offer an insight into how this differs from other financial products such as Forex or stock trading.

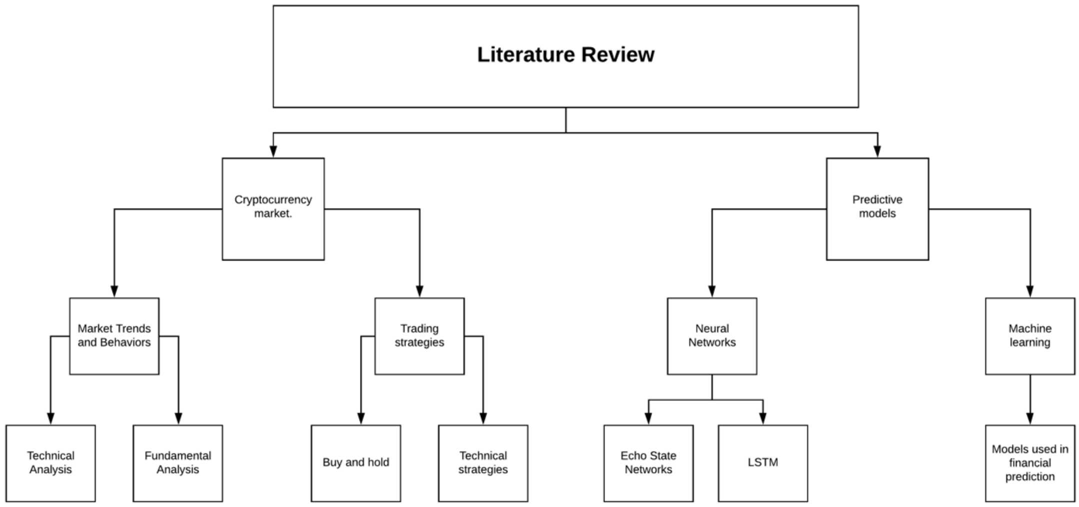


**Figure 2.1 Literature review chapter outline**

## 2.1 Cryptocurrencies and their place as a financial product

“Cryptocurrencies are digital financial assets, for which records and transfers of ownership are guaranteed by a cryptographic technology rather than a bank or other trusted third party. They can be viewed as financial assets because they bear some value for cryptocurrency holders, even though they represent no matching liability of any other party and are not backed by any physical asset of value (such as gold, for example, or the equipment stock of an enterprise)” (Raiborn & Sivitanides, 2015)

Cryptocurrencies are designed in such a way to secure them from being duplicated, the platform which facilities the transfer of these assets is the “blockchain”, a peer-to-peer secure digital ledger, which is encrypted in different languages per currency, for example, Bitcoin is SHA-256, whereas Ether uses ethash.

### 2.1.1 Financial markets and inefficiency

There is international debate as to what the fundamental value of cryptocurrencies is, within research by (Cheah & Fry, 2015) researchers conducts an empirical investigation into the fundamental value of Bitcoin, it provides evidence that the value is zero. The paper provides empirical evidence to address the existence of a substantial bubble component in the Bitcoin market. (Cheah & Fry, 2015) also, highlight the profound economic and societal issues with Bitcoin, this study highlights a perspective on feature engineering, by using the asset classes for ‘regular’ currencies to evaluate a cryptocurrency and shows that these technical indicators may not work on cryptocurrencies.

From the growth and developments of the cryptocurrency market and specifically of Bitcoin and Ether, it is obvious that these markets will remain volatile and that any technical predictive model needs fundamental analysis of the currencies to truly understand the impact of decisions made by the developers, from Ethers perspective, and to understand the effect of regulatory input, as seen with the price boom of Bitcoin in 2017. In 2017 the growth of Bitcoin can be seen to be caused by Bitcoin being declared a legal tender in Japan, also in this year, there was a large number of investors buying Bitcoin for portfolios (Gradwell, 2020).

### 2.1.2 Key players in the cryptocurrency market

The Covid-19 pandemic a large ‘Black Swan’ in the financial industry, this has affected all financial predictions so it even more interesting to cryptocurrencies now, due to there being no regulatory authority for cryptocurrencies, although they are at this point (July 2020) and a bull run, there is a real possibility of the value hitting 0, although unlikely due to the increase in “whales” large single holders of the currency/stock, there is more than any other tradable item, a possibility for its value to evaporate at any point.
The overall price increase in the last 5 years of Bitcoin and Ether have created a handful of millionaires, from early miners to investors such as The Winklevoss Twins (Tyler and Cameron), who claim to own approximately 1% of all Bitcoins in circulation, they are also the founders of Gemini, the world’s first regulated exchange for cryptocurrencies.

### 2.1.3 Market predictability

There is much speculation to the efficiency of the cryptocurrency market, the efficient market hypothesis as explained by (Malkiel, 2003), states that assets prices reflect all available information, with the concept that it is impossible to “beat the market” since the market only reacts to new information.

“Markets do not follow a random walk and are persistent, which is inconsistent with market efficiency”(Caporale et al., 2018), this makes predictive models easier, as the markets are not dependent on new variables to dictate their price, the influence of external factors are reduced.

(Kurihara & Fukushima, 2017) explore the market efficiency of Bitcoin, although their evidence shows the market is currently inefficient and that Bitcoin exhibits speculative bubble elements, it shows that Bitcoin transactions are becoming more efficient. comparatively, newer cryptocurrencies to the market do not yet show this inefficiency. A broader study looking at the efficiency in the market of cryptocurrencies (Tran & Leirvik, 2020), which reviews the top five cryptocurrencies, shows Litecoin to be the most efficient, and Ripple the least, with Bitcoin and Ether getting Adjusted Market Inefficiency Magnitude (AMIM) scores of 0.081 and 0.063 respectively, with 0 being the optimal score.

### 2.1.4 Comparison of Bitcoin and Ether

“Scarcity is a prerequisite for ascribing value to any form of money.” (Böhme et al., 2015), although cryptocurrencies are decentralized, mining is a key aspect to their supply.

Bitcoin and Ether are the two most dominant cryptocurrencies currently on the market, they have a combined market cap of over $247Billion, as of the end of July 2020.
The activity of the currencies can be seen in Table 2.1 Current Bitcoin and Ether activity. 30/08/2020.

Bitcoin and Ether are based on mining with ‘Proof of work’, Miners create new blocks in the chain by completing complex algorithms with large servers, these servers then store the transaction ledger for the currency, as a reward for this mining, the miner is awarded some of the tokens/coins of the currency.

Bitcoin was designed as a deflationary currency, to ensure it became scarcer over time. Bitcoin controls the flow of supply by having a maximum of 21million coins to ever be produced, it is predicted that it will take until 2140 for all to be mined, as although technology advances and the computation power to mine becomes more accessible, therefore every 210,000 blocks, which is approximately every 4 years, the block reward is halved. Block rewards started as 50 coins per block mined, and it currently stands at 6.25 coins per block, as per the ‘May-halving’ of 2020.

Unlike Bitcoin, Ether is an inflationary currency, it does therefore not have a halving event, but does reduce miners’ rewards over time, the developers of Ethereum plan to ditch the proof-of-work and move to a proof-of-stake where the network is secured by owners of the tokens and not by miners, this is commonly debated and discussed online, with the concept of Ethereum 2.0 being debated by the developers currently.

While, Bitcoin was created as an alternative to government-controlled currencies, and therefore was always aimed to be a currency of sorts, whereas Ether was intended to be a platform to facilities applications, smart contracts via the use of its own currency.

Ether, it is the native cryptocurrency of Ethereum, “the world’s programmable blockchain”, it was released in 2015, it utilizes block-chain technology not just as a decentralized payment network, but can be used to power decentralised contracts and applications. Decentralized Application (dapp) platform.
unlike Bitcoin, Ethereum is programmable, which means that developers can use it to build their own applications.

| Date: 30/08/2020 | Bitcoin | Ether |
|---|---|---|
| Market cap | $ 215,415,967,447 | $ 47,891,028,857 |
| Price | $11,659.77 | $426.09 |
| Volume(24h) | $18,898,773,498 | $10,536,235,593 |
| Circulating supply | 18,450,150 BTC | 112,397,729 ETH |
| Encryption algorithm used | SHA-256 | ethash |

**Table 2.1 Current Bitcoin and Ether activity. 30/08/2020**[2]

[2] https://coinmarketcap.com/

## 2.2 Technical and fundamental analysis in cryptocurrency

Technical analysis is the study of historical pricing to predict future pricing, whereas fundamental analysis looks at the fundamentals of an asset.

Fundamental analysis is the concept that if an asset has intrinsic value, identifying when that may be disproportionate to its current market values is when you would trade. It is based on this. Fundamental analysis is about doing your market research, looking outside of the previous pricing to analyse the market the asset is in and predicting its growth or potential losses from this.

Metcalfe's Law, states that "the value of a network is proportional to the square of the number of connected users of the system", this law shows a clear approach to fundamentally valuing crypto-assets. Fundamental indicators include transaction value, mining cost, unique addresses.

Technical analysis forces on former pricing and volume indicators of an asset, within this study, the focus will be on technical analysis, using technical indicators used in state-of-the-art studies, as outlined below.

### 2.2.1 Technical indicators and models

Technical indicators can provide a rich source of information for models, as seen in (Dai et al., 2012) who focus on the parameter selection of the Asian Stock markets, using their novel approach to combine nonlinear independent component analysis (NLICA) and neural networks, which outperform their baseline neural network. While, hybrid models such as (Zainuddin et al., 2019), demonstrate a novel hybridization of bootstrap and double bootstrap on Forex, which provided a higher accuracy, efficiency and precision. High-dimensional technical indicators have also shown results within predicting bitcoin returns by (Huang et al., 2019), who uses 124 technical indicators within a classification tree-based model.

While there are fundamental analysis models such as (Greaves & Au, 2015) who create classification model based on feature engineering from the Bitcoin transaction graph, this feature engineering technique can also be seen in (Dai et al., 2012).

The importance of technical indicator selection is very clear from the literature, with various approaches to the number of indicators required. (Huang et al., 2019) use 124 technical indicators, while (Lendasse et al., 2000) uses independent component analysis to provide a non-linear vectorized input. (Madan et al., 2015) investigates the Automation of Bitcoin Trading and only use 25 indicators. While (Lui et al., 2005) highlights the importance of the proper selection of input dimensions but also the time-delay between inputs.

## 2.3 Feature selection and model tools

Feature selection is critical when modelling for cryptocurrencies, due to the decentralized nature of the currencies, (Jang & Lee, 2018) highlight the issues in the volatility of Bitcoin, they examine the features from BlockChain information that is deeply involved in Bitcoin's supply and demand. Using these features aided them in predictions on a Bayesian Neural Network.

(Madan et al., 2015), compare an automated Bitcoin trading strategy and compare it to machine learning algorithms, using 25 features to predict the daily price change, they have a classification accuracy of 98.7%, from their binomial generalized linear model. The features which they use are both technical indicators and block-related inputs, such as transaction per block.

Feature selection ranges per research paper with (Greaves & Au, 2015) starting with 11 features but post feature pruning ending up with 7 features into the model, contrary to this (Sin & Wang, 2017) use 200 features of the cryptocurrencies used to feed into their ensembles of neural networks.

(Dutta et al., 2020), Plot 20 features and reveal that the endogenous features are more correlated with Bitcoin prices than the exogenous features – e.g. Google trends, interest-rates and Ripple prices are the most correlated exogenous. Variance inflation factor (VIF) is then calculated to reduce the collinearity of the features, which leads to 15 features to be added to the model.

This highlights the importance of feature selection within Bitcoin, which can be assumed to translate to Ether although separate EDA's will be completed on them, 80 technical indicators are chosen to fix to the data, an EDA will then be performed on these features, in order to complete feature pruning.

## 2.4 Predictive models

The use of neural networks is evident as a popular method within financial markets prediction, each model has a different activation function, the popular baseline is a traditional linear and non-linear approach compared to a dynamic approach.

Positive results can be seen with several models such as GARCH (Guresen et al., 2011) (Lendasse et al., 2000; Indera et al., 2017; Amirat & Zaidi, 2016) , non-linear approaches LSTM models (Madan et al., 2015; Sang & Di Pierro, 2019), SVM, (Chatzis et al., 2018; Madan et al., 2015; Nahil & Lyhyaoui, 2018) and Hybrid models using neural networks such as, (Jain & Kumar, 2007; Zainuddin et al., 2019) and a Bayesian-based model (Jang & Lee, 2018), another method widely used is machine learning classification, providing the direction of returns as (Enke & Thawornwong, 2005; M. Qiu & Song, 2016).

(Dutta et al., 2020), provides a Gated Recurrent Unit (GRU) approach to bitcoin, which had some feature engineering, which shows a promising financial gain.

Within this section of the literature review, there will be a focus on the neural network approach to the prediction of both cryptocurrencies and other financial products. Each type of neural network will briefly be discussed, with an in-depth review of novel techniques proposed.

## 2.5 Echo State Networks

Echo state networks are a type of recurrent network, they are chaotic in nature as they have random connections between the neurons, they were first proposed by (Jaeger & Haas, 2004), to learn nonlinear systems and predict chaotic time series.
"The core of the ESN is a large fixed reservoir. The reservoir contains a large number of randomly and sparsely connected neurons. Determination of the readout weights is the only trainable part, which can be obtained by simple linear regression", (Q. Li & Lin, 2016). The reservoir exhibits some special properties to decode the nonlinear dynamics well.

Echo state networks train by feeding the input forward, the neurons are updated for a while and observe the output over time. The input and output layers have an unconventional role, as the input layer is used to prime the network and the output layer acts as an observer of the activation patterns that unfold over time. During the training, only the connections between the observer and the hidden units are changed.

Echo state networks initialise connections within the neural network in such a way that there is a large reservoirs with coupled oscillators. By providing input to it converts the input to the state of the oscillators, the prediction is then based on the output of these oscillators. The unique element of echo state networks is that the network must only learn is how to couple the output to the oscillator, circumventing the need to learn the hidden to hidden connections or the input to hidden connections.

(Lin et al., 2009) investigate the use of an ESN in predicting stock market returns, using the Hurst's exponent to choose a persistent sub-series with the greatest predictability for training from the original set. A stock prediction system is built to forecast the next day closing price on stocks within the S&P 500. This study shows that ESN outperforms other neural networks in most cases. There were certain stocks which the ESN failed to predict, there the researchers applied PCA to filter noise and extract a reliable representation of the raw data, showing that a combination of PCA and parameter optimization increased the predictive power of the ESN.

### 2.5.1 Architecture and training algorithm

Training is based on not training the hidden to hidden at all, to fix the weights randomly and get them to learn sequences based on the effect on the output, this is similar to perceptrons.

Sensible sized random weights input, just learn the last layer, so that you are learning a linear model from the activities of the hidden units in the last layer as the output.

This increases the speed dramatically, as it is just learning a linear model. This relies on the idea that a big random expansion of the input vector to make it easy for the linear model to fit the data.

**Setting the random connection in an Echo State Network**

- Set the hidden -> hidden weights so that the length of the activity vector stays about the same after each iteration.
- Spectral radius is 1, or it would be 1 if it were a linear system.
- Use sparse connectivity – a few large weights, a lot of zero, therefore a lot of loosely coupled oscillator.
- Chose scale of input -> hidden connection; to drive the loosely coupled oscillators without wiping out the information from the past that they already contain.

The popularity of Echo state networks, within electrical systems and robotics, is due to the fact that they trained very quickly, as it is just a linear fit model, they demonstrate the importance of initial weight sensibility and are impressive modelling of one-dimensional time-series.

Key issues within Echo state networks are their need many more hidden units required than that required of an RNN.

## Equations of ESN

$$x(k+1)=\mathrm{sig}(\mathbf{W}_x \cdot x(k)+\mathbf{W}_{in} \cdot \mathbf{u}(k))$$

$$y(k) = \mathrm{w}^T \mathrm{x}(k)$$

**Equation 2.1: Dynamic and output equations of the ESN**

where $\mathbf{x}(k)$ is the reservoir internal state vector, $\mathbf{u}(k)$ and $y(k)$ are the input vector and the model output, respectively, sig denotes the sigmoid activation function, $\mathbf{W}_x$ denotes the internal connection weight matrix of the reservoir, $\mathbf{W}_{in}$ denotes the input weight matrix, and $\mathbf{w} = [\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_L]$ denotes the output weight vector, where $L$ is the size of the reservoir (the number of neurons in the reservoir).

The sole trainable part of the ESN is the output weight vector **w,** which can be determined by means of a simple linear regression

$$\boldsymbol{y} = \boldsymbol{\phi w} + \boldsymbol{\varepsilon}\mathbf{y}$$

**Equation 2.2: Simple linear regression**

Where

$$\boldsymbol{\phi} = [\mathbf{x}(k), \mathbf{x}(k+1), \ldots, \mathbf{x}(k+N-1)]^T$$

$$\mathbf{y} = [y(k), y(k+1), \ldots, y(k+N-1)]^T$$

**Equation 2.3 ESN output equation**

and $k$ is the beginning index of the training samples, which is usually set to discard the influence of the reservoir initial transient, $\varepsilon$ is assumed to be zero-mean Gaussian noise with variance $\beta$, and $N$ is the number of training samples.

The learning equations, in the state harvesting stage of the training, the ESN is driven by an input sequence, which yields an output sequence of extended system states. If the model includes output feedback (i.e., nonzero $\mathbf{W}_{fb}$), then during the generation of the system states, the correct outputs $\mathbf{d}(n)$ (part of the training data) are written into the output units ("teacher forcing"). The obtained extended system states are filed row-wise into a state collection matrix **S** of size $n_{max}$×(N+K). Usually, some initial portion of the states thus collected is discarded to accommodate for a washout of the arbitrary (random or zero) initial reservoir state needed at time 1. Likewise, the desired outputs **d**(n) are sorted row-wise into a teacher output collection matrix **D** of size $n_{max}$×L.

The desired output weights $\mathbf{W}_{out}$ are the linear regression weights of the desired outputs **d**(n) on the harvested extended states **z**(n). A mathematically straightforward way to compute $\mathbf{W}_{out}$ is to invoke the pseudoinverse (denoted by $\cdot^{\dagger}$) of **S** :

(3) $\mathbf{W}_{out}=(\mathbf{S}^{\dagger}\mathbf{D})$, (Jaeger 2003).

## 2.6 LSTM for financial modelling

Long/short term memory (LSTM) networks are a type of recurrent neural network, which attempt to combat the vanishing/exploding gradient problem by introducing gates and an explicitly defined memory cell. By truncating the gradient where this does not do harm, "LSTM can learn to bridge the minimal time lags by enforcing constant error flow through constant error carousels"(Hochreiter & Schmidhuber, 1997)

The LTSM neural networks provide with a robust extension of the recurrent neural network (RNN) topology in terms of nonlinear modelling and more importantly forecasting. In this regard, deep learning LTSM neural networks systems not only keep adjacent temporal information in a spontaneous manner but also control long-term (LT) information. Therefore, the LSTM can preserve previous information, which can significantly help improve its ability to learn signal sequences and inherent nonlinear patterns, such as those within cryptocurrencies. (Sang & Di Pierro, 2019)

Within LTSM is to introduce there are controlling gates, which control for the *input, forget* and *output* of each cell. The input gate determines how much current information should be treated as input to generate the current state, whilst the forget gate extracts how much information can be kept from the last prior state. The output gate filters the information that can be treated as significant and produces the output which basically in our context would be a forecast.

The three gates are set up with the following equations

$$\boldsymbol{i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)}$$

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$

$$o = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

**Equation 2.4 LSTM input, forget and output equation**

The tanh function which is used in(Lahmiri & Bekiros, 2020) will be used to process historical sequences as the inputs of the LSTM to extract hidden information, whereas the predicted digital currency price is regarded as the targeted output.

## 2.7 Algorithmic Trading strategies based on

The output of this research project if continued beyond its current scope would implement the predictive models into a trading strategy. Strategies which are used within by professional traders will have a fundamental and technical aspect to them. Although this is beyond the scope of this research, it is import to understand some of the most used trading strategies, in order to understand the approach this experiment is designed on.

Particularly within the feature selection of this experiment, many of the technical indicators are based on the below strategies. This is not a comprehensive list of trading strategies, simply an introduction to basic strategies, which the experimental models use some of the indicators and could be used in some strategies in future work.

### 2.7.1 Mean reversion strategies

Mean reversion strategies use the moving average as a technical analysis tool, the moving averages of a set number of days, it predicts the next day price, based on the average over the last number of days.
Other examples of mean reversion are pairs trading, selling options and using the CBOE Volatility Index.

### 2.7.2 Bollinger Bands

Bollinger bands are a trading tool which allows traders to determine the entry and exit points for a trade. The indicator focuses on price and volatility within the market. Within the calculation, there are three bands.
Bollinger bands use the moving average as the middle band, with the upper band using the middle band, plus twice the daily standard deviation, the lower band is the middle band, minus two standard deviations.

### 2.7.3 Moving average convergence divergence (MACD)

MACD is a trend following indicator that looks at a combination of two moving averages. Short-term moving average and Long-term moving average are set by the trader.
Both are combined to identify what is the current trend and if there is a change in the momentum, used to identify if the market is bullish or bearish.

### 2.7.4 Relative Strength Index

Relative strength index (RSI) is used as an indicator of temporarily overbought or oversold market conditions. RSI is widely used an as technical indicator and an oscillatory. When the RSI value is over 70, it indicators that the product is overbought when it is under 30 it indicates that the market is undersold.

### 2.7.5 Stochastic oscillator trading strategy

The stochastic oscillator is a momentum indicator comparing the closing price of a security to the range of its prices over a certain period. The sensitivity of the oscillator to market movements is reducible by adjusting that time period or by taking a moving average of the result.

### 2.7.6 Momentum

Profit from a continuation of a certain move, this can be seen to be widely used by traders following Bitcoin price surge in 2017, the increase in investment in Alt-coins such as ETH and XRP, as their prices also increased.

Examples of momentum strategies include, Gap and go strategy (if stock gaps up by X% overnight then it will go up), Earnings (bet on a continuation of a price move – coupled with a Gap & Go), Sector momentum and Break out and break down strategy.

### 2.7.7 Sentiment

Peoples current opinion and attitude towards given security and generates a market assumption based on these results.

Usually using millions of data sources to create these algorithms, may go through for example tweets and classify whether the consensus is positive or negative.

## 2.8 Overview

This chapter discussed cryptocurrencies and provided an in-depth analysis on research within the market, not only in machine learning but in economic factors which affect cryptocurrencies. The market predictability of cryptocurrencies can be seen to be examined by several economists, we will investigate the claims that the market is inconsistent with market efficiency.

Section 2.1.4 which explores the similarities and differences of Bitcoin provide insight into the fundamental differences, although on exploration of the data, it will be interesting to examine the correlation between the markets prices, and if the differences truly have an impact on the overall market.

It is clear from the examination of several predictive models that this is an area of great interest both by researchers, financial traders and economists, although this paper is only investigating the use of time-series prediction models, the context provided by this chapter will lead to the formation of the parameters and features used within those models.

Table 2.2 Summary of models used, summarizes the key models used in academic papers which are presented in this literature review. Although there are other references and the models will be influenced by all references, Table 2.2 provides a visual summary of papers which focus on time-series modelling.

| | Generalized regression | Echo State | LSTM | Linear regression | GRU | XGBoost – regression trees | Neural network | SVM - regressor | Classification | Genetic Algorithm | Bayesian | GARCH | RBF NN | MSE | RMSE | Accuracy | Sharpe ratio | feature engineering |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (Adachi & Aihara, 1997) | x | | | | | | | | | | | | | | | | | |
| (Alessandretti et al., 2018) | | | x | | | x | | | | | | | | | | | x | |
| (Chatzis et al., 2018) | | | | | | x | | | | | | | | | x | | | |
| (D. Li et al., 2012) | | x | | | | | | | | | | | | | x | | | |
| (Dai et al., 2012) | | | | | | | x | | | | | | | | | | | x |
| (Dutta et al., 2020a) | | | x | | x | | | | | | | | | | x | | | x |
| (Enke & Thawornwong, 2005) | | | | x | | | x | | | | | | | x | | | | |
| (Greaves & Au, n.d.) | | | | x | | | | x | x | | | | | | | | | x |
| (Guresen et al., 2011) | | | | | | | | | | | | x | | x | | | | |
| (Huang et al., 2019) | | | | | | | | | x | | | | | | | | | x |
| (Jaeger, 2004) | | x | | | | | | | | | | | | x | | | | |
| (Jain & Kumar, 2007) | | | | | | | x | | | x | | | | | | | | |
| (Jang & Lee, 2018) | | | | x | | | | x | | | x | | | | x | x | | x |
| (Lahmiri & Bekiros, 2019) | x | | x | | | | x | | | | | | | | x | | | |
| (Lee, 2019) | x | | | | | | | | | | | | | x | | x | | |
| (Lendasse et al., 2000) | x | | | x | | | | | | | | | x | | | | | |
| (Li et al., 2013) | | | | | | | | | | x | | | | | | | | |
| (Lin et al., 2009) | | x | | | | | | | | | | | | | | | | |
| (Lui et al., 2005) | | | | | | | | | | | | | | | x | x | | x |
| (M. Qiu & Song, 2016) | | | | | | | | | | x | | | | | | | | x |
| (Madan et al., n.d.) | | | | | | | | | x | | | | | | | | | x |
| (Nahil & Lyhyaoui, 2018) | | | | | | | | x | | | | | | | | | | x |
| (Ning et al., 2009) | x | | | | | | | | | | | | | | | | | |
| (Q. Li & Lin, 2016) | | | | | | | x | | | | | x | | | x | | | |
| (Sang & Di Pierro, 2019) | | | | | | | x | | x | | | | | x | | x | | |
| (Sin & Wang, 2017) | x | | | | | | | | | x | | | | x | | | | x |
| (Skowronski & Harris, 2006) | | x | | | | | | | | | | | | | | | | |
| (Y. Qiu & Lee, 2019) | x | | | | | | | x | | | | | | x | | | | x |

**Table 2.2 Summary of models used**

# 3. DESIGN AND METHODOLOGY

This chapter presents the experimental design and methodology used. It describes the processes, names the critical tools deployed for analysis and explains the main aim of each aspect of the experiment.

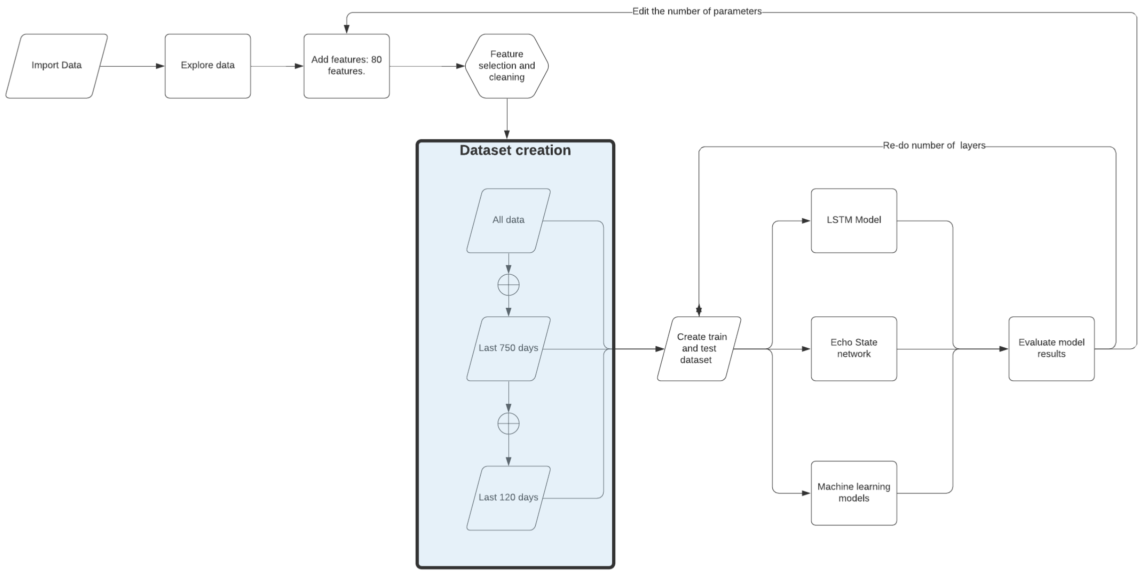


**Figure 3.1 Experimental design**

This study will present findings from several predictive models will be created, a comparison of these models will be made using MSE and an accuracy calculation $R^2$, on two major cryptocurrencies, Bitcoin and Ether.

This experiment was conducted using Python, the data for this experiment was sourced from Yahoo finance and all packages and libraries will be cited in attached code submissions.

There are two currencies being examined within this experiment and each will go through the same processing, within feature selection, the datasets will become unique to the currency.

### 3.1.1 Experimental environment

Data pre-processing:

- Key packages: NumPy, pandas, matplotlib

Data processing

- Key packages: TA-lib, eli5,

Model constructions:

- Echo State Network
    - Key packages: pyESN
- LSTM
    - Key packages: TensorFlow
- Pipelines created with sklearn
    - Linear Regression
    - Bayesian Ridge
    - ElasticNetCV
    - Random Forest Regressor – using n_estimator of 500
    - XgBoost Regression – extreme gradient boosting
    - Neural Network
        - Activation function – 'ReLU'
        - Optimizer – 'adam'

Model evaluation

System used: Colab by Google

## 3.2 Data Preparation

This section of the chapter will discuss the cryptocurrency selection process, and analyse the input variables of the stock, to conduct this research the daily historical value of the cryptocurrencies were selected.

The historical data is collected from Yahoo Finance for each currency consists of the daily features: Open, Close, High, Low, Adjusted Close and Volume. Each of these features is used to apply 80 of the most popular technical indicators, which will then be explored and the cross-validation of the most key variables using eli5 will be saved for prediction. These will be examined further and used within the machine learning models.

### 3.2.1 Data Processing

Add features

- List in Technical indicators: 80 features, Table 3.1: Features added, shows the codes and descriptions of features added, for those with calculations over periods, example EMA, the value of 5, 14, 30 days are inserted and all are added to the dataset.

Feature selection

- A random forest regressor calculation is complete, to determine if the model is overfitting.
- If so, Cross-validate the technical indicators to delete some of the noise using permutation importance, with the $R^2$ value set as the calculation to maximize. The permutation importance calculation is complete with the eli5 package and iterated through the random forest model, dropping the features and evaluating their weighted importance into calculating the highest $R^2$ and then ranking their importance.
- Any negative permutation score indicates that the feature negatively affects the prediction score and is therefore removed.

| | TA-Lib code | Description |
|---|---|---|
| Overlapping | BBAND WIDTH, BBAND UPPER SIGNAL | Bollinger Bands |
| | EMA | Exponential Moving Average |
| | DEMA | Double Exponential Moving Average |
| | MIDPRICE | Midpoint Price over period |
| | SAR | Parabolic SAR |
| | SMA3, SMA5, SMA10, SMA20 | Simple Moving Average |
| | TEMA | triple exponential moving average |
| | TRIMA | Triangular Moving Average |
| | WMA | Weighted Moving Average |
| Momentum indicators | ADX14, ADX20 | Average Directional Movement Index |
| | APO | Absolute Price Oscillator |
| | DX | Directional Movement Index |
| | MACD, MACDSIGNAL, MACDHIST | Moving Average Convergence/Divergence |
| | MINUS_DI | Minus Directional Indicator |
| | MINUS_DM | Minus Directional Movement |
| | MOM1, MOM3, MOM5, MOM10 | Momentum |
| | PLUS DI | Plus Directional Indicator |
| | PLUS DM | Plus Directional Movement |
| | PPO | Percentage Price Oscillator |
| | ROC | Rate of change |
| | ROCP | Rate of change Percentage |
| | ROCR | Rate of change ratio |
| | ROCR100 | Rate of change ratio 100 scale |
| | RSI5, RSI10, RSI14 | Relative Strength Index. |
| | SLOWK, SLOWD | Stochastic |
| | FASTK, FASTD | Stochastic Fast |
| | TRIX | 1-day Rate-Of-Change (ROC) of a Triple Smooth EMA |
| Volatility indicators | ATR | Average True Range |
| | NATR | Normalized Average True Range |
| | TRANGE | True Range |
| Pattern recognition | CDLSHORTLINE | Short Line Candle |
| | CDLTASUKIGAP | Tasuki Gap |
| Cycle indicators | HT DCPERIOD | Hilbert Transform - Dominant Cycle Period HT |

**Table 3.1: Features added**

Clean dataset saved

- Within the models – clean dataset uploaded
- Scale data using Robust scaler from the sklearn package
- Convert the data frame to scaled array
- Splitting the data – will be described per model.

## 3.3 Testing for chaotic non-stationary elements

### 3.3.1 Lyapunov exponent

Lyapunov exponent illustrates the bounded dynamical systems sensitivity to initial conditions(Cosme Andrieu & Steeb, 2005), here the positive Lyapunov exponent indicates chaos and unpredictability, the algorithm used in calculating the Lyapunov exponent is (Rosenstein et al., 1993) which estimates the largest Lyapunov exponent.

All dynamical systems having at least a positive exponent is defined as being chaotic, and that "the magnitude of this exponent reflects the scale of time on which this system becomes unpredictable".(Zerroug et al., 2013)

One of the efficient methods consists to measure the average exponential rate of divergence/convergence of neighbour orbits in a phase space, Equation 3.1 is for one-dimensional discrete dynamical system, $x_{k+1} = f(x_k)$, with the initial condition $x_0$, the Lyapunov exponent is defined as:

$$\boldsymbol{\lambda}(\mathbf{x_0}) = \lim_{n\to\infty} \sum_{k=1}^{N} lg|f'(x_k)| = \lim_{n\to\infty} \frac{1}{N} \lg\left(\prod_{k=1}^{N} |f'(xk)|\right)$$

**Equation 3.1 Lyapunov exponent**

For a finite amount of time, Lyapunov transforms to:

$$\boldsymbol{\lambda}(\mathbf{x_0}) = \frac{1}{N} \sum_{k=1}^{N} lg|f'(x_k)|$$

**Equation 3.2 Lyapunov exponent - finite time**

### 3.3.2 Hursts exponent

The Hurst exponent, proposed by (Hurst, 1951) in a study on the use of long-term storage reservoirs. Within this study the Hurst's exponent was developed for use in fractal analysis, to provide a measure for the long-term memory and fractality of a time series. Hursts exponent is a mean reversion calculation, it assists in determining whether a time series is a random walk (H ~0.5), trending (H >0.5) or mean-reverting (H <0.5) for a specific time period. Hurst exponent, as used by (Carbone et al., 2004) in forecasting price returns and volatility, highlights the importance of the datasets stability for predictions.

The Hurst exponent is defined as

$$\boldsymbol{(R|S)_n = \frac{1}{k}\sum\nolimits_{j=1}^{k}\left[\frac{R_j(t)}{S_j(t)}\right] = cn^H}$$

**Equation 3.3 Hursts exponent**

Where H represents the Hursts exponent, c is a constant, $S_j(t)$ is the standard deviation of the sub-time series.

### 3.3.3 Detrended fluctuation analysis

Detrended fluctuation analysis is a method for determining the statistical self-affinities of a signal, DFA can be used for non-stationary processes whose mean and variance change over time.

In order to calculate the DFA, the algorithm converts the bounded time-series into an unbounded process X, to calculate the cumulative sum $X_t$ , then $X_t$ is divided into time windows of length n, and each window is locally tested for the least-squares straight line fit. $Y_t$ is the result of the piece-wise sequence of the straight-line fits.

The root-mean-square deviation from the trend is the fluctuation which is calculated as:

$$\boldsymbol{F(n) = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(X_t - Y_t)^2}}$$

**Equation 3.4 Detrended fluctuation analysis**

### 3.3.4 Dickey-Fuller test

The Dickey full test is used to determine the presence of unit root in the series and therefore understand whether the series is stationary or not.

The Null Hypothesis: The series has a unit root (value of a = 1)

Alternative Hypothesis: The series has no unit root.

## 3.4 Evaluation metrics and early stopping

The loss function used within each experiment is the mean squared error function, which measures the average of the squares of the errors.

### 3.4.1 Mean Squared Error

Mean squared error measures the average of the squared of the errors, it is used to measure the difference between values predicted by the model and the values observed.

$$\boldsymbol{MSE = \frac{1}{n}\sum_{i-1}^{n}\left(Y_i - \hat{Y}\right)^2}$$

**Equation 3.5 Mean-squared error equation**

Where N is the number of observations used for testing, Y is the true value, $\hat{Y}$ is the forecasted value and **T** is time script.

### 3.4.2 Early stopping

Early stopping is a form of regularization, it is used within the model to prevent overfitting.

Early stopping will be used with TensorFlow models, where the validation loss minimum is the target variable and is measured per iteration (epoch), experimentation to find the most suitable patience number is completed.

Through experimentation, it is found that a patience of 20 epochs, meaning the value cannot have grown over any of the previous 19 epochs is set. The model is given a 500 epoch range. Provides the best results.

## 3.5 Models standard

Each model will produce a loss metric as MSE and an accuracy metric of $R^2$, which will be used to compare the models. The data imported to each model will go through a Robust Scaler.

## 3.6 Machine learning models

Pipelines are created within the sklearn library to create 6 regression models.

- Linear Regression
- Bayesian Ridge
- ElasticNetCV
- Random Forest Regressor
- XgBoost regression
- Neural Network with ReLU

## 3.7 LSTM models

Long-short term memory models, with the tanh function, optimizer Adam, batch size of 128 and a validation set of 10% will be used for all LSTM models.

<u>The activation function:</u>

The tanh function is defined as:

$$\tanh(x) = \frac{2}{1+e^{-2x}} - 1$$

This activation function was chosen as it is nonlinear in nature, so there can be stacked layer, there is a bounded range of (-1,1), although a distinct issue with the tanh activation function is the vanishing gradient problem. The reason this is chosen over ReLU is that the data is highly fluctuating and with ReLU is the so-called “dying ReLU”, where if a neuron gets negative it is unlikely to recover. Another issue with ReLU which is not with tanh is the large outputs and the likelihood to explode. Therefore tanh is used.

<u>The optimizer</u>

The optimizer which will be used for all LSTM models is Adam, which is the adaptive moment estimation. This optimizer is used for its computational efficiency, it is appropriate for non-stationary variables with a noisy and sparse gradient.

<u>Batch and validation size</u>

The number of examples per batch is set to 128, with iterations being set at early stopping, the same batch size was used across all experiments to ensure a level of stability within the models.

Validation size was set to 10%, as a standard validation size.

## 3.8 Echo State Network

The echo state network is a type of recurrent neural network which are part of the reservoir computing framework, the hidden layer within the network is considered the reservoir.

Due to the chaotic nature of the Echo state network, several parameter optimization activities will be completed on the network to examine its usefulness in predicting a chaotic-time series.

Within this study, the ESN used will be from the pyESN library (Korndörfer, 2015).

The aspects of the network are the

- Input weights and the reservoir are randomly assigned and are not trainable.
- The weights of the output neurons are trainable.
- The reservoir is sparsely connected, which ensures it does not overtrain
- The only weights trained are the output weights to the output layer
- The output layer is a linear layer, which performs linear regression.
- Training complexity is of the order O(N), where N is the number of hidden units in the reservoir.

Aim:

The aim is to predict the short (1-day), medium (5-day) and long (30-day) closing prices of the input data.

Experimentation:

The adjusted parameters will be the hyperparameters of sparsity, spectral radius and noise will be tuned to produce the best prediction, as measured by the least mean squared error.

The dataset will learn from 1500 previous days, to predict out the short, medium and long term, using a validation set of 120 days.

| | | |
|---|---|---|
| N_inputs | Number of input dimension | Fixed to one |
| N_outputs | Number of output dimensions | Fixed to one |
| N_reservoirs | Number of hidden/reservoir neurons | Fixed to: 500 |
| Sparsity | Proportion of recurrent weights set to zero | 0.2, 0.4, 0.6,0.8 |
| Spectral radius | Spectral radius of the recurrent weight matrix | 0.5, 0.7, 0.9, 1, 1.1, 1.3, 1.5 |
| Noise | Noise added to each neuron (regularization) | 0.0001, 0.0003,0.0007, 0.001, 0.003, 0.005, 0.007,0.01 |
| MSE | Mean square error | Output of models |

**Table 3.2 ESN experimental design**

| | | |
|---|---|---|
| Input_shift | Scalar or vector of length n_units added to each input dimension before feeding it into models | none |
| Input scaling | Scalar or vector of length n_inputs to multiply with each input dimension before feeding it into the network | none |
| Teacher forcing | If true, feed the target back into the output units | True |
| Teacher_scaling | Applied to the target signal | None |
| Teacher_shift | Additive term applied to the target signal | None |
| Out activation | Output activation function | Linear |
| Inverse out activation | Inverse Output activation function | Identity |
| Silent | Suppress messages | True |

**Table 3.3 ESN set features**

**Initialise recurrent weights.**

- Begin with a random matrix centred around zero
- Delete the fraction of connection given by sparsity
- Compute the spectral radius of these weights
- Rescale them to reach the requested spectral radius

**Input weights**

- Random input weights
- Random feedback(teacher forcing) weights

**Next step**

The network then updates itself, by performing one update steps, where it computes the next network state by applying the recurrent weights to the last state and feeding in the current input and output patterns.

**Fitting the network**

The network will collect the reaction to the training data, train readout weights.

Inputs into the model are : N_training_samples x n_inputs

Outputs: N_training samples x n_outputs.

The network will then return an output on the training data using the trained weights.

**Predictions from network**

Apply the learned weights to the network's reaction to new input. the network will start from the last training state.

# 4. IMPLEMENTATION AND RESULTS

This chapter will review the implementation of the experiment described in Chapter 3, the sections of this chapter will include, data preparation and exploration, modelling and a comparative evaluation of the models.

Within each section associated with modelling a recap of the experimental design used, the testing and training results and iterations, the results for that model and an evaluation of the model and conclude with a comparative evaluation of the models.

## 4.1 Protocol of experiments

- Import data from Yahoo finance.
- Add technical indicators to the data
- Perform feature importance on the data and clean the dataset based on this
- Load the data into models
- Analyse results of the models
- Tune hyperparameters
- Re-run the models
- Compare the model performance with other models

## 4.2 Data preparation

This experiment focuses on two different cryptocurrencies, the data will be imported for both currencies within the same time period, it will then be explored visually and statistically.

Data is imported from Yahoo finance, the variables retrieved are: open, high, low, close, adjusted close and volume. The library Technical Analysis Library (TA-lib) is then used to add 80 technical features onto the data, these technical analysis terms are chosen as they mirror the there is no external data added into the dataset, to ensure that calculations are solely completed on a technical level. The data will then undergo feature selection, here the features will be selected based on the currencies themselves and a new database will be formed including these features.

## 4.3 Data Exploration

Bitcoin and Ether are both openly traded cryptocurrencies, as the value of each is drastically different, the chosen visualization is the cumulative return, as up until April of 2017, ether was valued below $50 the scale is changed to May 2017 – August 2020.

Ether was first released on the 30th July 2015 initially until May 2017, it was in the early mining and valuation phase and therefore provides a start image of returns, therefore the graph of cumulative return is taken from August 2017 to August 2020.

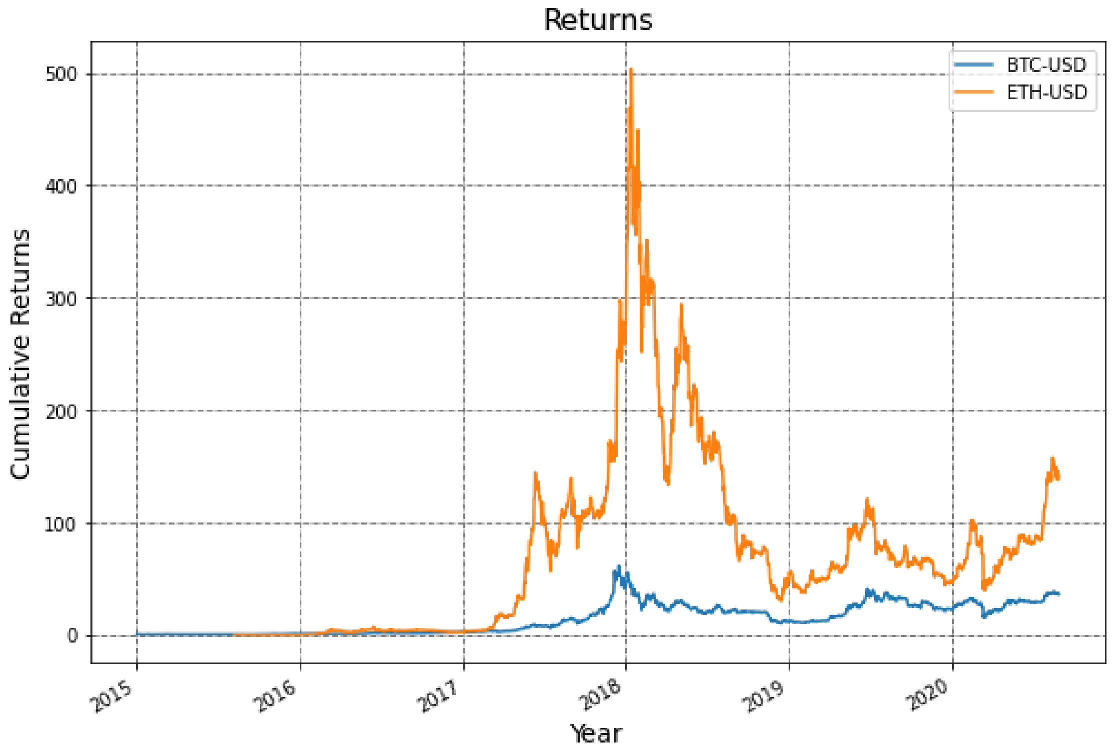


**Figure 4.1 Cumulative returns from 01/01/2015 to 28/08/2020**

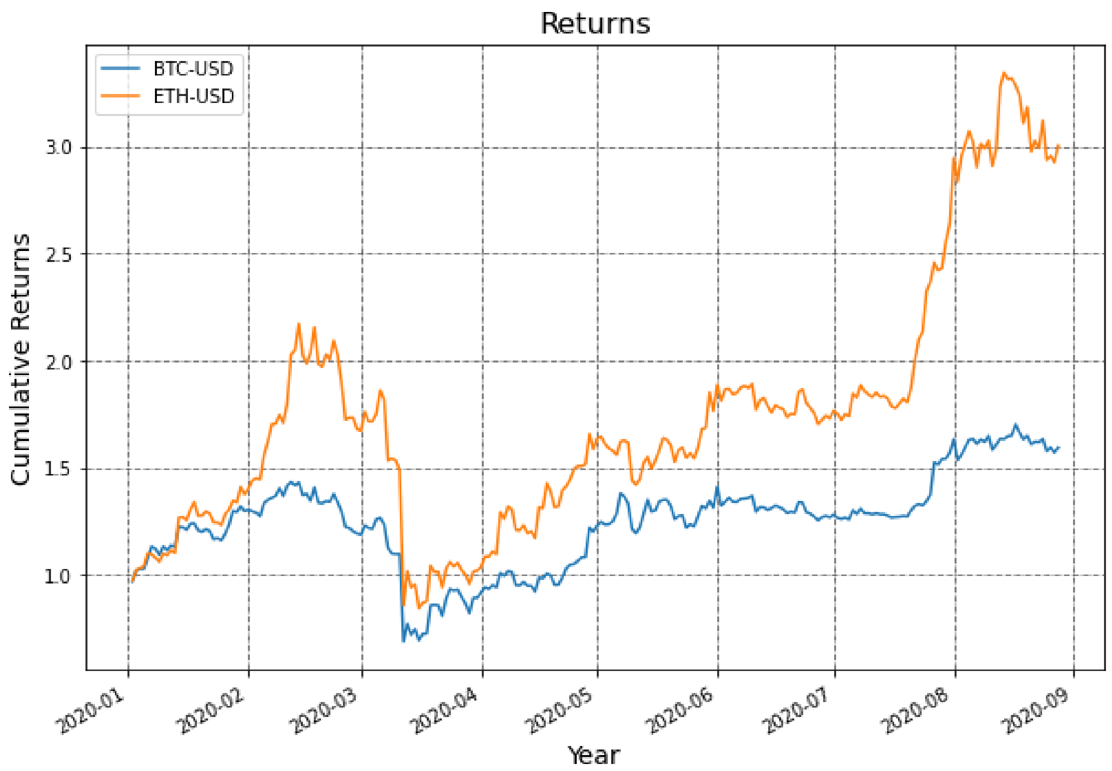


**Figure 4.2 Cumulative returns from 01/01/2020 to 24/08/2020**

| **28/08/2020** | **BTC-USD** | **ETH-USD** |
|---|---|---|
| count | 1123 | 1123 |
| mean | 7776.79847 | 316.434031 |
| std | 2796.19794 | 227.310543 |
| min | 2710.66992 | 84.3082962 |
| 25% | 6159.80493 | 172.647476 |
| 50% | 7678.24023 | 229.668045 |
| 75% | 9415.84619 | 380.597504 |
| max | 19497.4004 | 1396.42004 |

**Table 4.1 Data description from 01/08/2017 – 28/08/2020**

## 4.4 Testing for chaotic, non-stationary elements

Tests conducted on the datasets to examine the chaotic, non-stationary elements are as described in the experiment design chapter.

They are

- Lyapunov exponent
- Hursts exponent
- Detrended fluctuation analysis
- Dickey-fuller analysis

### 4.4.1 Results and discussion of non-linearity investigation

Results of the explained test are in Table 4.2 Non-linear evaluation of datasets, calculations are complete on the entire dataset, the last 748 days and the last 120 days for both Bitcoin and Ether. The data investigated is the closing price of the currencies.

We can see that the data is chaotic in nature with a positive Lyapunov, but that the data is strong trending with a Hurst value (H >0.5) for each of the datasets. A DFA calculation is performed to confirm that the underlying process is non-stationary, which is true for (DFA >1).

The dickey-fuller test is also complete, due to the (p> 0.05) we can see that the dataset is non-stationary and indicates non-stationary data.

The Hurst exponent for BTC and ETH is reduced for the last 120-day dataset, comparatively the p-value is at its lowest for the whole dataset.

The results from Table 4.2 Non-linear evaluation of datasets indicate that the data is non-stationary and chaotic in nature. The chaotic nature of the dataset has not stabilised over several time periods and even with looking back only 120 days, the data still remains chaotic in nature.

| | BTC (1760) | BTC (748) | BTC (120) | ETH (1760) | ETH (748) | ETH (120) |
|---|---|---|---|---|---|---|
| Start date | 2015-10-27 | 2018-08-06 | 2020-04-27 | 2015-10-27 | 2018-08-06 | 2020-04-27 |
| End date | 2020-08-25 | 2020-08-25 | 2020-08-25 | 2020-08-25 | 2020-08-25 | 2020-08-25 |
| Days in set | 1760 | 748 | 120 | 1760 | 748 | 120 |
| Lyapunov | 0.005549 | -0.00309 | 0.05567 | 0.000884 | 0.01498414 | 0.048502 |
| Hurst | 0.9235399 | 0.921658 | 0.8494935 | 0.90994571 | 0.927042608 | 0.88588865 |
| DFA | 1.5761045 | 1.5677553 | 1.464936 | 1.6101971 | 1.56893227 | 1.6524709 |
| Dickey fuller stat | -1.85169 | -1.1669 | -1.0324 | -2.4916 | -1.9336 | -1.2597 |
| p-value | 0.355118 | 0.687683 | 0.741256 | 0.117516 | 0.316421 | 0.931014 |

**Table 4.2 Non-linear evaluation of datasets**

## 4.5 Transforming the data to be stationary

The scope of this experiment is to predict chaotic non-stationary data, therefore we will visually investigate methods to transform the data into a stationery set, but will use the Robust Scaler and technical indicators previously discussed for the implementation.

In order to make the time-series appropriate for a lot of predictive models, it must have stationary data, therefore seasonality is tested for, this does not appear to be a viable method, therefore a log transformation is used.
Transformation is used to stabilize the non-constant variance of a series, a log transform is used and produces the results.
It can be seen from Figure 4.3 Log transform of closing price [BTC, ETH], that there are outliers present in the data, particularly of note is the effect of the March 2020 global pandemic.

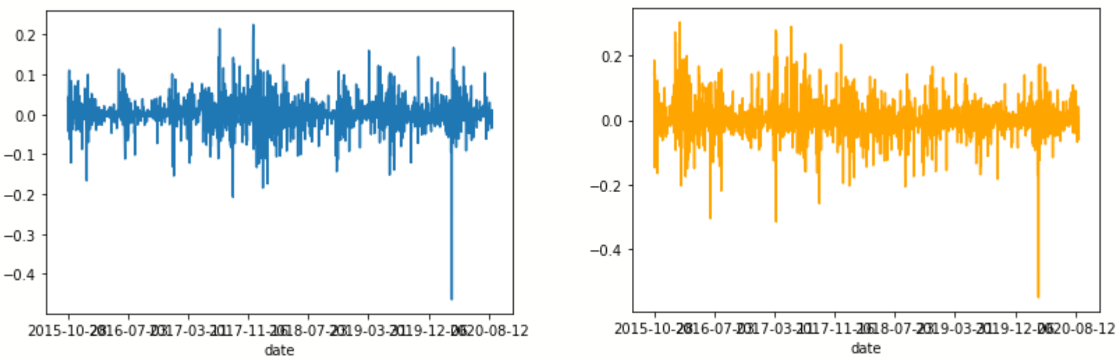


**Figure 4.3 Log transform of closing price [BTC, ETH]**

### 4.5.1 Rolling mean smoothing

Rolling means smoothing was attempted on the data, as can be seen with Figure 4.4 Rolling means smoothing (5-day) [BTC, ETH], this was not smoothing out the data, due to the significant chaotic nature and the rise and fall in prices in 2017, this method was not considered appropriate.
Using a 5-day and 30-day smoothing average technique, it can be seen that the data is still very chaotic with large peaks for both currencies.

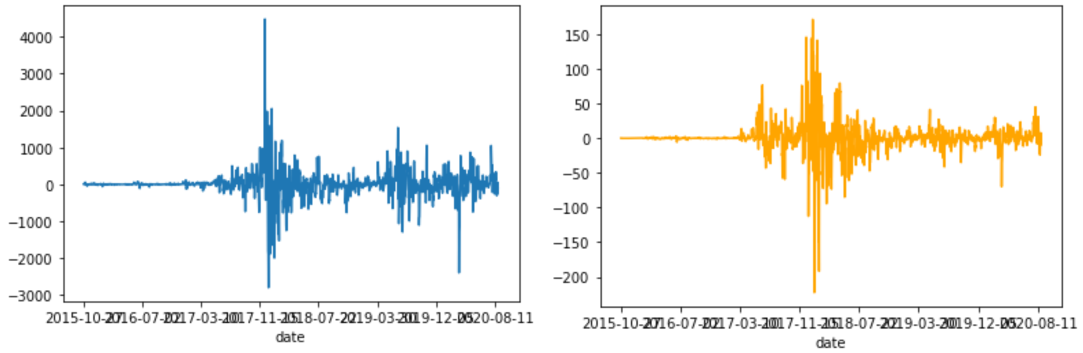


**Figure 4.4 Rolling means smoothing (5-day) [BTC, ETH]**

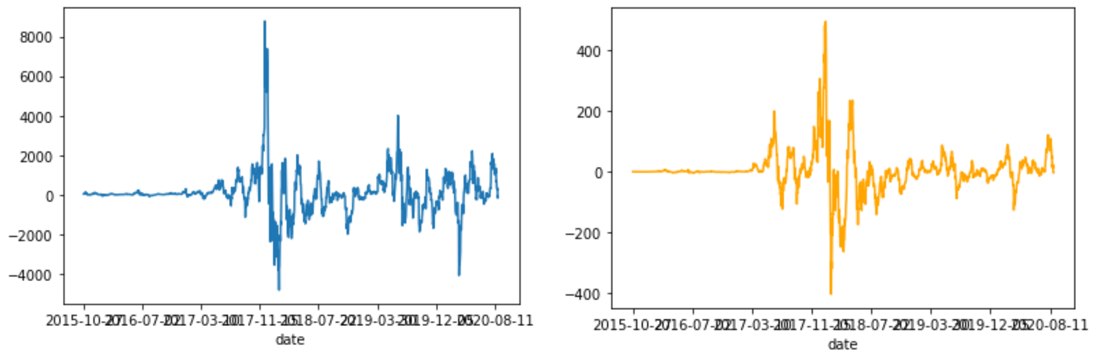


**Figure 4.5 Rolling mean smoothing (30-day) [BTC, ETH]**

### 4.5.2 Seasonality of the cryptocurrencies

Initially, the two cryptocurrencies are analysed, the correlation analysis finds that there is a Pearson's correlation coefficient of 0.91 between BTC and ETH.

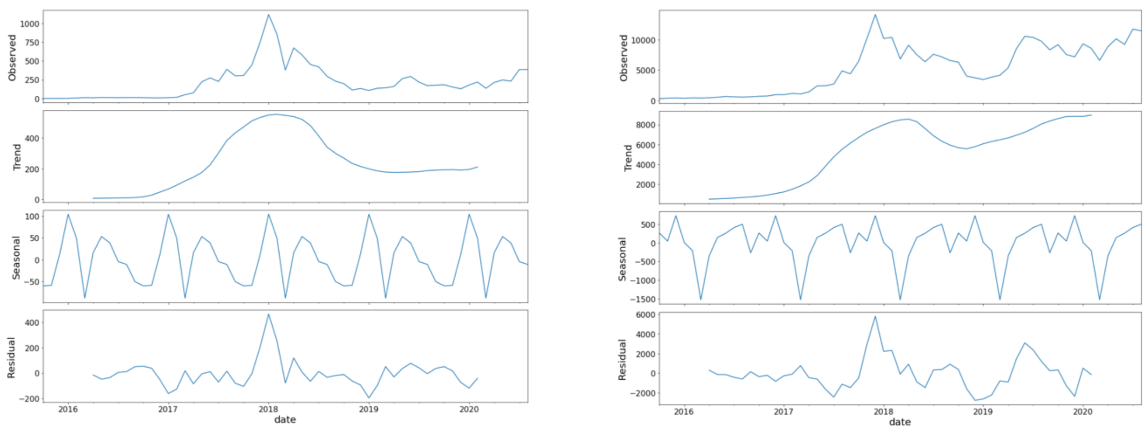


**Figure 4.6 BTC- ETH seasonality**

### 4.5.3 Evaluation of stationary cryptocurrency data

It can be seen from Figure 4.4 Rolling means smoothing (5-day) [BTC, ETH] and Figure 4.5 Rolling mean smoothing (30-day) [BTC, ETH], that the rolling mean is not an appropriate method to transform the dataset, this is due to how rapidly cryptocurrencies can change within one month. As an example, on 20th March 2020 Bitcoin was worth $6,483.74, by 20$^{th}$ April the value has risen to $8,773.11.

Therefore, it was decided that the data would be converted with the Robust Scaler, with different sets being learned are [120, 750, 1500]

## 4.6 Technical indicators; Feature importance

Within the data preparation phase, 80 technical indicators are added to both of the datasets, to ensure there is a reduced amount of noise within the data set feature importance analysis is complete.

With 'close price' as the main factor, correlation analysis was complete to reduce the number of variables that would be entering the model and therefore reduce noise.

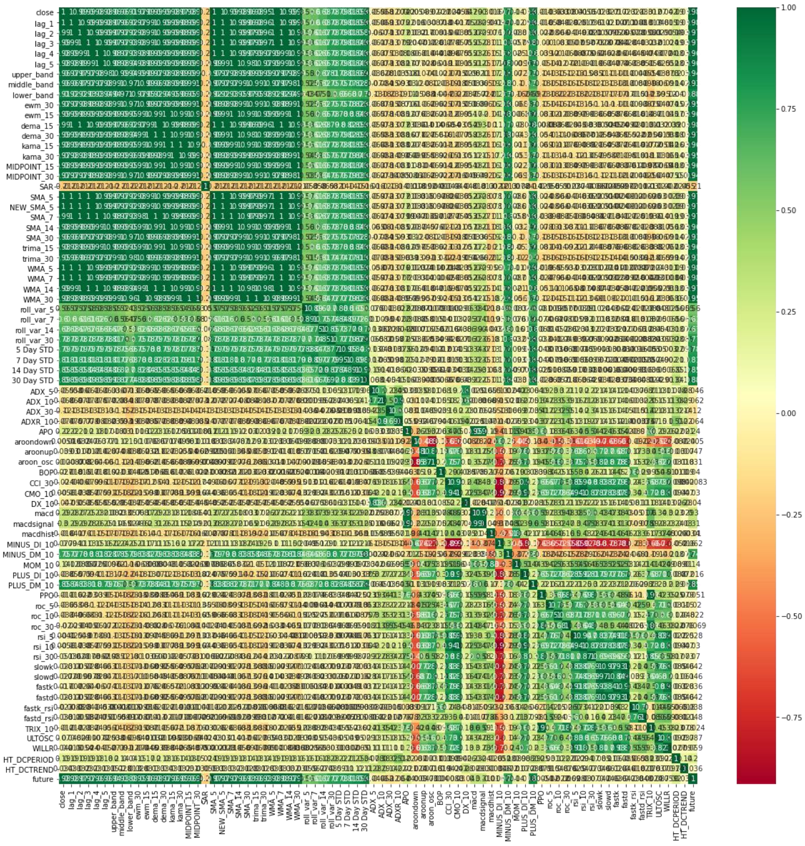

**Figure 4.7 Correlation matrix of 80 TA: BTC**

### 4.6.1 Permutation Importance

Permutation importance is a method to compute feature importance, it measures how the score decreases when a feature is not available, the method is also known as "mean decrease accuracy (MDA).

The $R^2$ score is used, with the dependent variables, all of the technical indicators and close price and the independent variable as the next-day closing price. A random forest regressor is used as the predictive model.

After running permutation importance from the rfpimp library on the data, with the future_close as the dependent variable, the results of which are presented in Figure 4.8 Results of permutation importance on TA, using elif, which uses cross-validation.

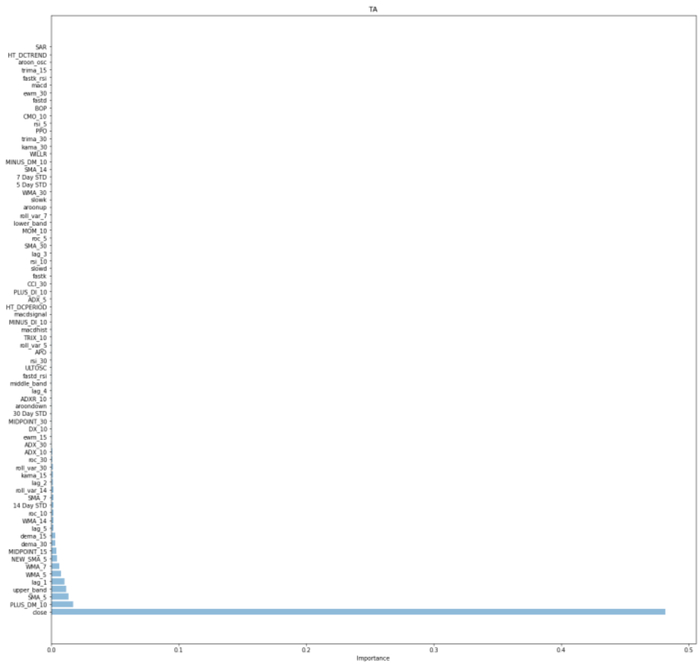


**Figure 4.8 Results of permutation importance on TA**

### 4.6.2 Feature selection

Using a random forest regressor model as a baseline to analyse the impact of features on the model produces interesting results, in Figure 4.9 ETH Column feature importance shows the negative performance features. Negative importance in this instance means that removing a given feature from the model will improve its performance. Although it mentions close, which is the highest correlated variable, this is due to the dependence of closing being so high, therefore it can be seen to be overdependent, but it will remain in the dataset as an independent variable while the other negative importance indicators will be removed.

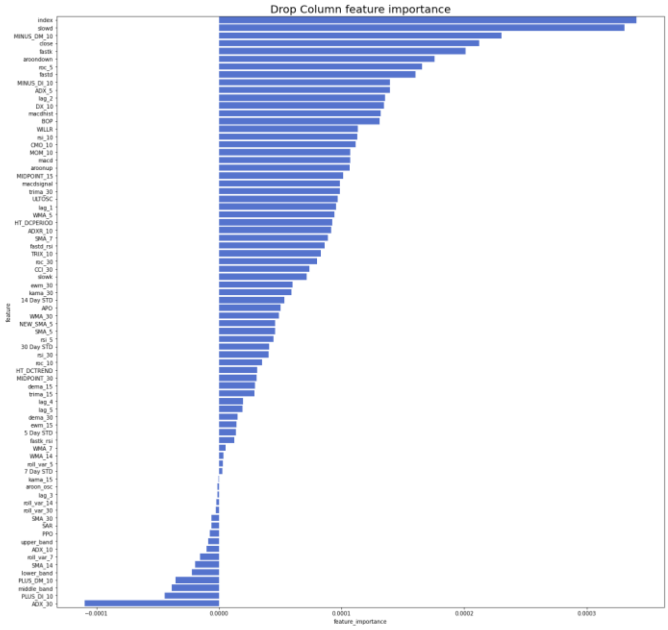


**Figure 4.9 ETH Column feature importance**

### 4.6.3 Technical indicators used

Resulting from the feature selection, the database for BTC and ETH supply different indicators, as well as a varying amount of importance on the features, it can be seen in Table 4.3 Feature importance rank table, that for BTC, the top-ranking indicator, other than closing, is the simple moving average of 5 days, whereas this ranked 56 for Ether, which ranked moving averages convergence divergence signal as its most important feature.

The features in Table 4.3 Feature importance rank table, are the features which will be used in the machine learning and TensorFlow model, these features will not be used in the Echo State network, due to its strength as a one-dimensional modelling tool.

| Indicator | Description | BTC - rank | ETH - rank |
|---|---|---|---|
| Std_Dev | Standard deviations, of [5, 7, 14, 30] days | [38,33,44,43] | [38,35, 19, 20] |
| ADX | The average directional movement over [5, 10, 30] days | [- ,- , 57] | [2, 50, 8] |
| ADXR_10 | Average directional movement index rating over 10 days | [51] | [11] |
| APO | Absolute price oscillator | 45 | 16 |
| aroon_osc | Aroon Oscillator, [overall, upward and downward] motion | [13 , 36, 15] | [9, 7,5] |
| BOP | Balance of power | 47 | 3 |
| CCI_30 | Commodity Channel index | 7 | 6 |
| CMO_10 | Change Momentum Oscillator | 9 | 4 |
| DEMA | Double Exponential Moving Average [15, 30 day] | [56, 55] | - |
| DX_10 | Directional Movement Index | 52 | 15 |
| EWM | Exponentially weighted moving average [15 days] | 53 | - |
| slow/fast[d, k] | Stochastic oscillators | [37, 50] | [33,39] |
| fastd_rsi | Stochastic RSI | [5,14] | [31,36] |
| HT_DCPERIOD | Hilbert Transform - Dominant Cycle Period | - | 45 |
| Kama | Kaufman Adaptive Moving Average (30) | 54 | - |
| lag_3 | Closing from 3 days previous | - | 46 |
| Lower_B | Bollinger bands - lower band | - | 47 |
| MACD | Moving Average Convergence Divergence | 12 | 10 |
| MACD_hist | Moving average history over the slow period of 30 days with a signal period of 5 days. | 49 | 21 |
| MACD_signal | Directional signal of MACD | 20 | 1 |
| Midpoint | Midpoint over period. [30 days] | 28 | 52 |
| MINUS_DI_10 | Minus Directional Indicator | 26 | 26 |
| MINUS_DM_10 | Minus Directional Movement | 23 | 12 |
| MOM_10 | Momentum | 8 | 17 |
| PLUS_DI_10 | Plus Directional Indicator | 11 | 43 |
| PLUS_DM_10 | Plus Directional Movement | 2 | 18 |
| PPO | Percentage Price Oscillator | 6 | 27 |
| ROC | Rate of change : | [21, 18, 25] | [32, 14, 24] |
| Roll_var | Rolling variable of [5, 7, 14, 30]days | [41, 40, 18, 25] | [49, 44, 41, 37] |
| RSI | Relative Strength Index [5, 10, 30] | [31, 46, 34] | [22, 13, 28] |
| SMA | Simple Moving Average [5,7,14,30] | [1, 48,32,30] | [56,59,- ,55] |
| TRIMA | Triangular Moving Average over 30 days | 29 | - |
| TRIX | 1-day Rate-Of-Change (ROC) of a Triple Smooth EMA of 10 days | 16 | 40 |
| ULTOSC | Ultimate Oscillator | 19 | 25 |
| WILLR | Williams' %R | 23 | - |
| WMA | Weighted moving average | [16, 29, 17, 29] | [53, 54, 58, 48] |

**Table 4.3 Feature importance rank table**

## 4.7 Splitting data

Several different models will be analysed in this experiment, machine learning models, LSTM model and an Echo State Network model, below will describe how the data is read into each model

### 4.7.1 Sklearn models data

Data will be read into Sklearn models as is, with Robust scaling.

Using the train_test_split algorithm, the data is split on 70% train, 30% test, with no validation and no shuffle.

The model will learn from looking back a set number of periods [all, 748, 120] and predicting out [1, 5, 30]

### 4.7.2 Echo state network model data

Data will be read into the ESN model only the ‘closing price’ with the date as the index.

The data will be scaled using a Robust Scaler with the range negative one to one.

The model will learn from looking back a set number of periods [all, 748, 120] and predicting out [1, 5, 30].

### 4.7.3 LSTM model

Data will be read into the LSTM model with Robust scaling, the scaler will be fit to the ‘closing price’ variable, and to the rest of the dataset separately, so that the inverse can be completed on the close column on output.

The future_close column will be the dependent variable.

The model will learn from looking back a set number of periods [all, 748, 120] and predicting out [1, 5, 30]

The data will be split into split-sequence windows of the size of the periods set, in order to learn as much from the data as possible.

## 4.8 Machine learning models

Machine learning without recurrent learning or backtesting can offer good predictions when the series is chaotic and non-stationary. Due to the Hursts exponent value above 0.5, the data can be considered to be trending, therefore some basic regression models are considered to evaluate the necessity of using complex deep-learning models, rather than machine learning models.

For the machine learning inspection, several pipelines are created with the following algorithms.

- Linear Regression
- Bayesian Ridge – Approach in which statistical analysis is undertaken with the context of Bayesian inference
- ElasticNetCV – regularised regression method that combines L1 and L2 penalties of the lasso and ridge methods
- Random Forest Regressor – using n_estimator of 500 – An Ensemble learning method, that constructs a constructs a multitude of decision trees at training time and outputting the mean prediction.
- XgBoost Regression – decision-tree based ensemble ML often using unstructured data
- Neural Network
    - Activation function – 'ReLU'
    - Optimizer – 'adam'
    - Hidden layers (8,8,8)
    - Max iterations: 500

### 4.8.1 Experimental design

Pipelines are an efficient and effective method within sklearn to build models quickly and reliably, therefore pipelines are created for each algorithm, base them all on MAE and accuracy score functions which are in sklearn.

All use Robust Scaler and feature ranking with recursive feature elimination.

Evaluation metrics

Error: Mean Absolute Error

Score: Accuracy ($R^2$)

### 4.8.2 Testing and training

Each model is run for short(1 day), medium(5 days) and long(30 days) returns of Bitcoin, Ether.

Training data is 70% of the entire data with 30% for testing.

### 4.8.3 Results

| | **Test** | **BTC_MAE** | **BTC_MSE** | **BTC_RMSE** | **BTC_Score** |
|---|---|---|---|---|---|
| 1 day | Linear regression | 224.445225 | 130703.372 | 361.529213 | 0.94585718 |
| | Bayesian Ridge | 235.626552 | 131372.13 | 362.452936 | 0.94558015 |
| | Elastic Net CV | 533.21515 | 419161.78 | 647.427049 | 0.8263656 |
| | Random Forest | 263.323552 | 156112.453 | 395.110685 | 0.93533168 |
| | XGBoost regression | 309.834892 | 212341.594 | 460.805376 | 0.91203921 |
| | NN regression | 1821.77864 | 5683007.05 | 2383.90584 | -1.3541398 |
| 5 day | Linear regression | 707.0275 | 880569.888 | 938.386854 | 0.63682719 |
| | Bayesian Ridge | 699.328325 | 884022.489 | 940.224701 | 0.63540324 |
| | Elastic Net CV | 693.468395 | 792622.69 | 890.293598 | 0.67309919 |
| | Random Forest | 719.96298 | 870510.72 | 933.01164 | 0.64097589 |
| | XGBoost regression | 686.251131 | 863182.218 | 929.076002 | 0.64399838 |
| | NN regression | 1777.81479 | 4719417.23 | 2172.42197 | -0.9464259 |
| 30 day | Linear regression | 2023.01004 | 5773491.37 | 2402.80906 | -1.308744 |
| | Bayesian Ridge | 1888.14072 | 5256411.69 | 2292.68657 | -1.1019706 |
| | Elastic Net CV | 1743.14349 | 4577157.47 | 2139.42924 | -0.8303457 |
| | Random Forest | 1841.44096 | 4563210.46 | 2136.16724 | -0.8247684 |
| | XGBoost regression | 1884.94446 | 5512659.52 | 2347.90535 | -1.2044408 |
| | NN regression | 3531.84716 | 17536760.2 | 4187.69151 | -6.012722 |

**Table 4.4 Results of Machine learning algorithms on BTC**

| | **Test** | **ETH_MAE** | **ETH_MSE** | **ETH_RMSE** | **ETH_Score** |
|---|---|---|---|---|---|
| 1 day | Linear regression | 8.66028628 | 142.942923 | 11.955874 | 0.96438601 |
| | Bayesian Ridge | 8.32292916 | 134.954367 | 11.6169862 | 0.96637634 |
| | Elastic Net CV | 9.40391584 | 166.846236 | 12.9168973 | 0.95843054 |
| | Random Forest | 10.1825122 | 185.578851 | 13.6227329 | 0.95376334 |
| | XGBoost regression | 8.45081407 | 141.49437 | 11.8951406 | 0.96474691 |
| | NN regression | 21.9298978 | 736.447774 | 27.1375713 | 0.81651525 |
| 5 day | Linear regression | 24.8698287 | 1014.39813 | 31.8496174 | 0.74710152 |
| | Bayesian Ridge | 24.1752812 | 947.927764 | 30.7884356 | 0.76367317 |
| | Elastic Net CV | 17.9524301 | 625.42739 | 25.0085463 | 0.84407539 |
| | Random Forest | 35.5438642 | 1878.67054 | 43.3436333 | 0.53163072 |
| | XGBoost regression | 20.2874591 | 709.258898 | 26.631915 | 0.82317545 |
| | NN regression | 43.5962501 | 2914.10098 | 53.9824137 | 0.27348869 |
| 30 day | Linear regression | 74.8671487 | 9596.05138 | 97.9594374 | -1.3997474 |
| | Bayesian Ridge | 63.4674759 | 7611.21358 | 87.2422695 | -0.903386 |
| | Elastic Net CV | 47.3348164 | 4621.68295 | 67.9829607 | -0.1557745 |
| | Random Forest | 61.5572659 | 6260.32978 | 79.1222458 | -0.5655617 |
| | XGBoost regression | 61.6486964 | 6234.01511 | 78.9557794 | -0.558981 |
| | NN regression | 60.5324499 | 5869.79016 | 76.6145558 | -0.4678969 |

**Table 4.5 Results of Machine learning algorithms on ETH**

### 4.8.4 Evaluation

The machine learning models show promising results for 1 day and 5-day models, with the neural networks being the most underperforming models of each dataset.

The random forest regressor performs with the highest accuracy of each grouping.

Due to the Hursts exponent of (H > 0.5) meaning trending, it is understandable that the next day predictions will have a low MSE and a high R2 score. What is surprising is the high MSE score for the neural networks at each point, but due to lack of deep learning and the high trend exponent, it is not surprising that the other models are performing well.

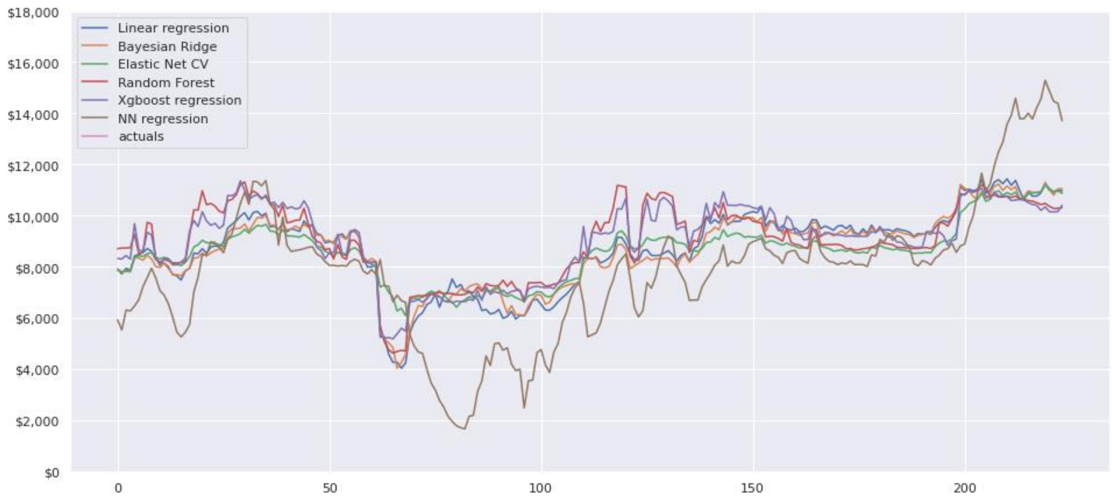


**Figure 4.10 BTC machine learning performance over training set**

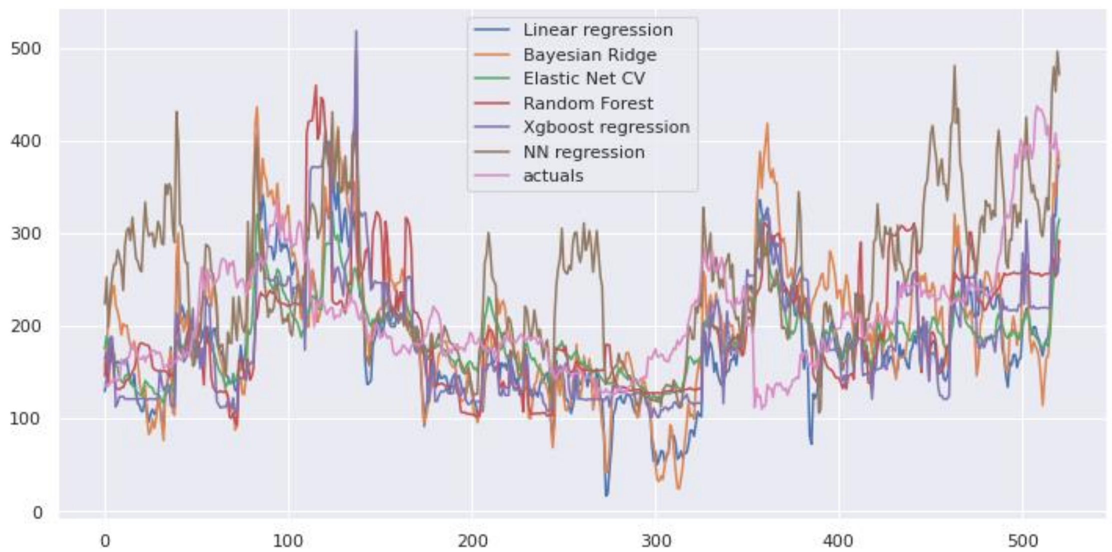


**Figure 4.11 ETH machine learning performance over training set (5-day)**

## 4.9 Echo state model

The Echo state model is created with library pyESN.

### 4.9.1 Experimental design

Data is imported and converted with the RobustScaler.

The model will be initially run with a reasonable parameter for sparsity, spectral radius and noise, a hyperparameter optimization will then be run to determine the spectral radius and noise which produces the lowest MSE and highest $R^2$ score, as per Table 4.6 The design of the experiment.

Then the model will be run with the optimized hyperparameters.

Results will then be presented and analysed.

| | | |
|---|---|---|
| N_inputs | Number of input dimension | Fixed to one |
| N_outputs | Number of output dimensions | Fixed to one |
| N_reservoirs | Number of hidden/reservoir neurons | Fixed to: 500 |
| Sparsity | Proportion of recurrent weights set to zero | 0.2 – to increase the chaotic nature of the model and ensure no overfitting. |
| Spectral radius | Spectral radius of the recurrent weight matrix | [0.5, 0.7, 0.9, 1, 1.1, 1.3, 1.5] |
| Noise | Noise added to each neuron (regularization) | [0.0001, 0.0003,0.0007, 0.001, 0.003, 0.005, 0.007,0.01] |
| MSE | Mean square error | Output of models |

**Table 4.6 The design of the experiment**

### 4.9.2 Hyperparameter optimization

Parameters radius and noise are investigated per each of the short, medium and long term prediction states. Sparsity factor which adds to the chaotic neuter. of this neural network, is tested on the models to test MSE per model.

| ETH | Noise | Spectrum radius | MSE |
|---|---|---|---|
| Pre-Optimization | 0.0003 | 0.5 | 0.003513044 |
| Post- optimization | 0.0003 | 1.3 | 0.002228509 |

**Table 4.7 Pre and post optimization results**

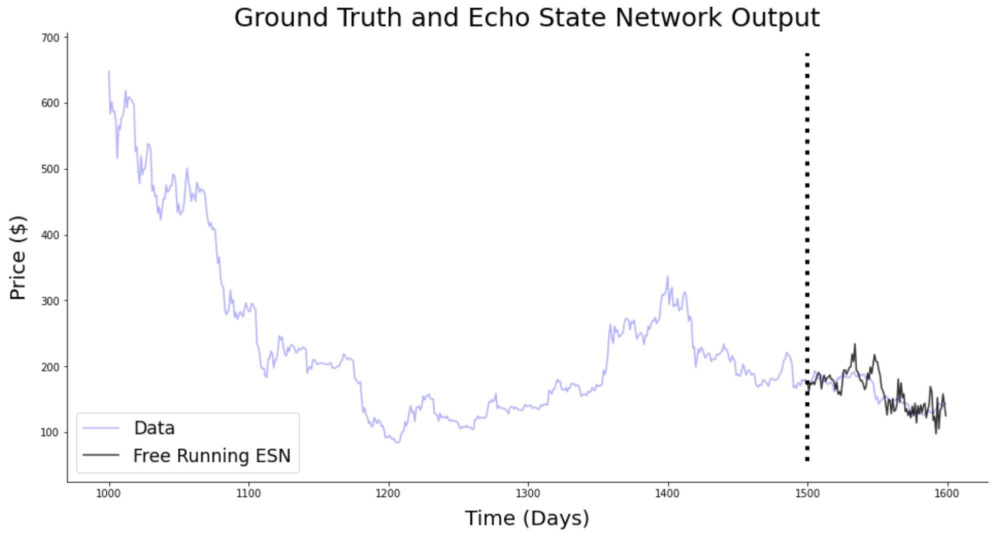


**Figure 4.12 ETH prediction Pre-optimization**

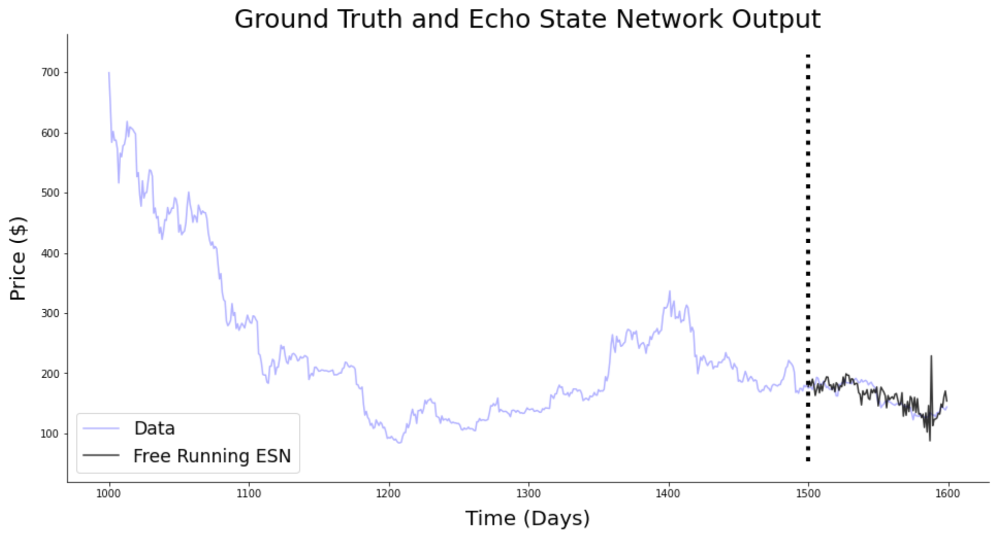


**Figure 4.13: ETH prediction Pre-optimization**

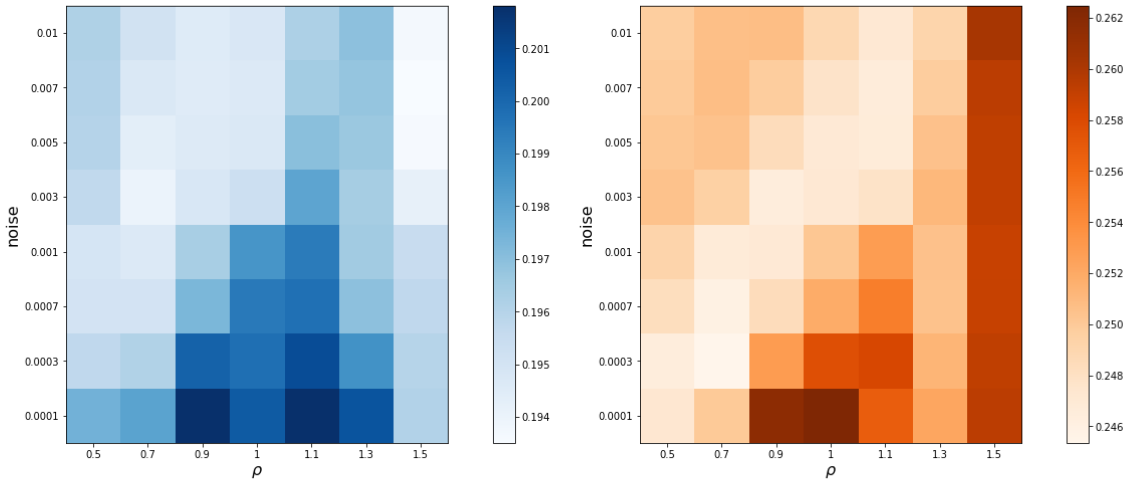


**Figure 4.14 BTC & ETH spectral radius and noise parameters**

| | N_res | sparsity | spectrum radius | noise | MSE | $R^2$ |
|---|---|---|---|---|---|---|
| ETH parameters | 500 | 0.2 | 0.7 | 0.0003 | 0.26247969 | 0.8517659 |
| BTC parameters | 500 | 0.2 | 1.5 | 0.007 | 0.19807439 | 0.7712362 |

**Table 4.8 Results of the hyperparameter optimization**

### 4.9.3 Results

Initially, the model is run with the hyperparameter optimization being on reducing the Mean Squared Error (MSE), as can be seen in Figure 4.15 MSE as a Function of Window Length, the MSE rises over the window length with the 1 day out being the parameter, therefore hyperparameters were optimized to the window length as well as the sparsity, spectral radius and noise.

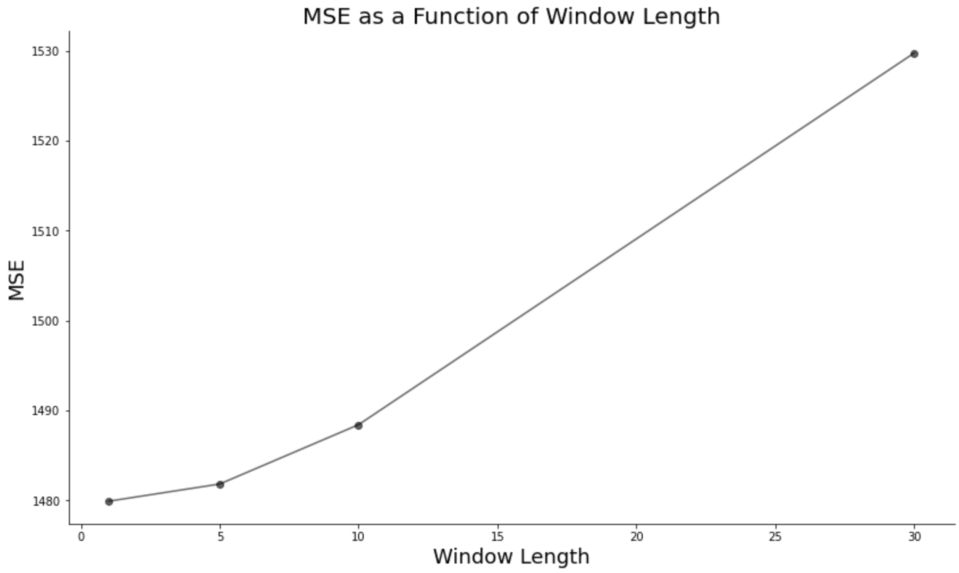


**Figure 4.15 MSE as a Function of Window Length**

The results in Table 4.9 ESN results table, shows the results of the hyperparameter optimization for both Bitcoin and Ether, showing the clear difference between the parameters for the both currencies and how the effect of the different parameters has on the $R^2$ result and MSE.

| | Days | Learn | Test | Nodes | Sparsity | Radius | Noise | MSE | R2 |
|---|---|---|---|---|---|---|---|---|---|
| BTC | 1 | 1500 | 120 | 1000 | 0.4 | 1.5 | 0.007 | 0.20752 | 0.66943 |
| | 1 | 1500 | 120 | 1000 | 0.4 | 1.5 | 0.007 | 0.269255 | 0.777983 |
| | 5 | 1500 | 120 | 1000 | 0.4 | 1.5 | 0.007 | 0.293504 | 0.276596 |
| | 30 | 1500 | 120 | 1000 | 0.4 | 1.5 | 0.007 | 0.718219 | -0.17164 |
| | 1 | 1500 | 120 | 1000 | 0.2 | 1.5 | 0.007 | 0.258795 | 0.745553 |
| | 5 | 1500 | 120 | 1000 | 0.2 | 1.5 | 0.007 | 0.267426 | 0.663679 |
| | 30 | 1500 | 120 | 1000 | 0.2 | 1.5 | 0.007 | 0.281068 | -1.9449 |
| ETH | 1 | 1500 | 10 | 500 | 0.4 | 1.2 | 0.0001 | 0.255008 | 0.03999 |
| | 1 | 1500 | 100 | 500 | 0.4 | 1.2 | 0.0001 | 0.255008 | 0.896397 |
| | 1 | 1500 | 100 | 1000 | 0.4 | 1.2 | 0.0001 | 0.259065 | 0.680415 |
| | 1 | 1500 | 120 | 1000 | 0.2 | 0.7 | 0.003 | 0.25512 | 0.87057 |
| | 5 | 1500 | 120 | 1000 | 0.2 | 0.7 | 0.003 | 0.265371 | 0.726895 |
| | 30 | 1500 | 120 | 1000 | 0.2 | 0.7 | 0.003 | 0.270608 | -0.77257 |

**Table 4.9 ESN results table**

### 4.9.4 Prediction of models

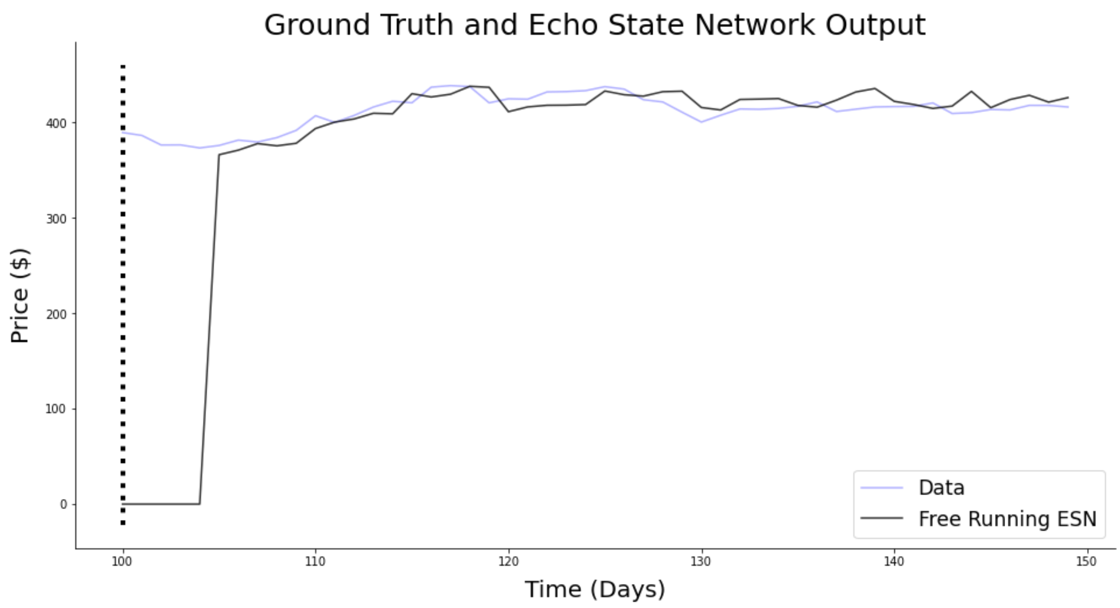


**Figure 4.16 ETH predictions Learning days 100**

### 4.9.5 Evaluation

It can be seen from the results that there is a high MSE associated with Echo state networks, due to the data being non-stationary and highly chaotic, from the Hurst exponent (H>0.5) and p-value (p>0.05) for each split of the data, it is not surprising that the Echo state networks provide a poor prediction result.

We can see the positive effects of parameter optimization within Table 4.7 Pre and post optimization results, with these hyperparameters then placed into the model, there is a positive result showing increased $R^2$ than when random parameters were chosen.

The key feature of Echo State Networks is following patterns, since both cryptocurrencies are currently unstable and relatively new, this method is not currently an appropriate method for predicting these two cryptocurrencies. Although we see results improve with shorter-term learning rate, it may provide much higher MSE than those seen in machine learning results.

It can also be seen from Figure 4.16 , that using the log_diff still had inaccurate results, from this figure it can also be seen that the direction is often incorrect.

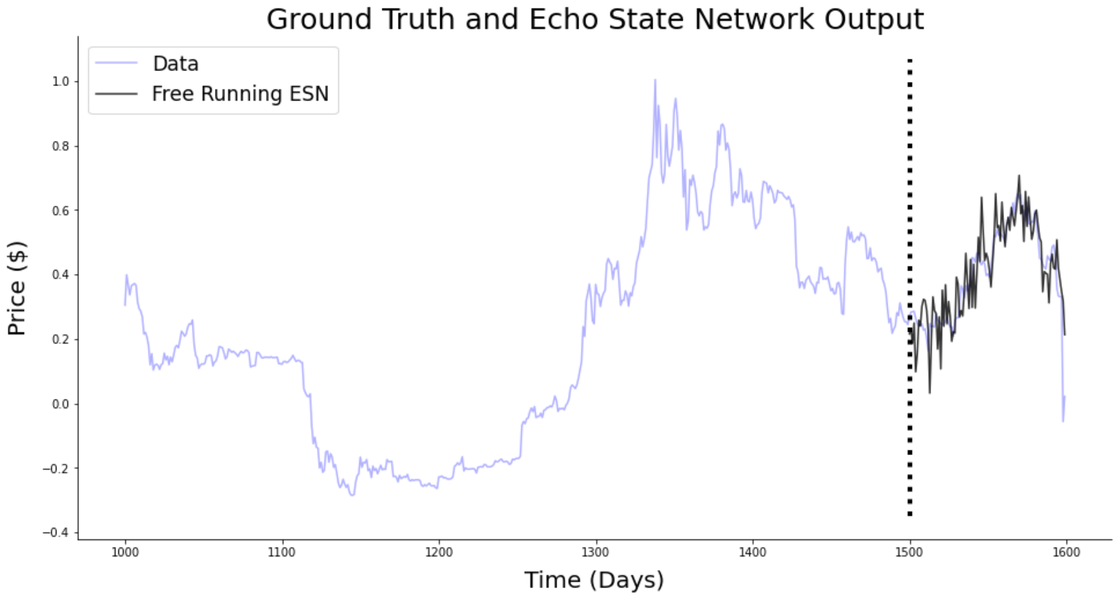


**Figure 4.17 BTC: highest R2 graphed**

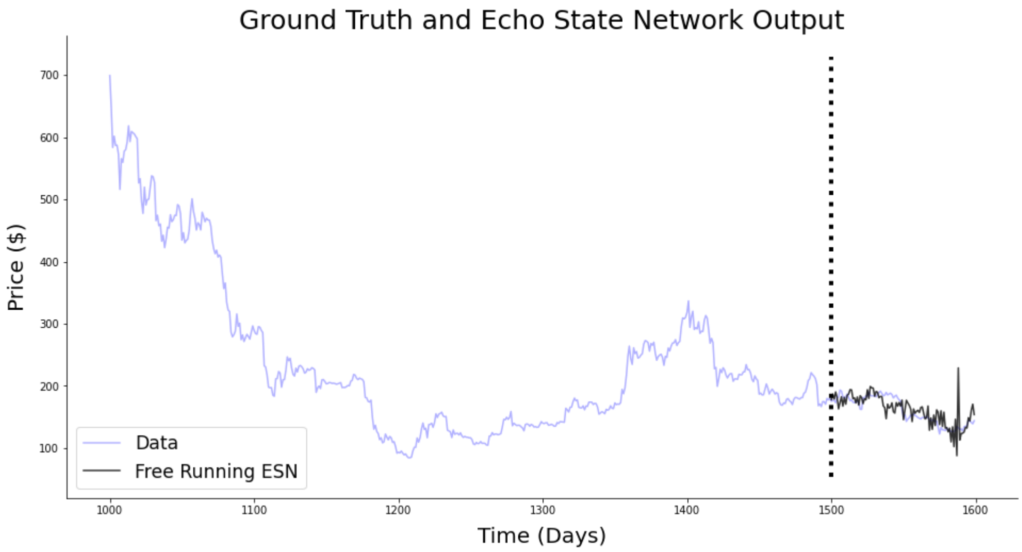


**Figure 4.18 ETH: highest R2 graph**

## 4.10 LSTM network

An LSTM is another type of neural network, here TensorFlow with Keras is used as the wrapper for the model.

### 4.10.1 Experimental design

The data is read into the model and then a Robust scaler is applied to the ‘close price’ and the rest of the dataset separately so that the ‘close price’ can quickly be inversed for graphs later.

Functions are created to run the model, a split_sequence function which will use the number of steps into the future to predict and the number of steps to learn from to produce windows of the data.

A visualization function is created to visualize the training of the data over each epoch, visualisations of the loss and accuracy of the training and testing data, provide insight into the models performance and its ability to predict new data through the epochs.

A layer maker function is also created, the number of layers can easily be adjusted, by calling this function, the inputs required are the number of layers that are being called to add, the number of nodes in the layers, the activation function to be used and the dropout rate.

A validator function is created to create prediction values for every interval, this will then be used to assist in creating a future prediction for the currencies.

A validation mean-squared-error function is created, to calculate the MSE between the prediction and actual data frames.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 365, 90)           54000
_________________________________________________________________
lstm_1 (LSTM)                (None, 365, 30)           14520
_________________________________________________________________
lstm_2 (LSTM)                (None, 365, 30)           7320
_________________________________________________________________
lstm_3 (LSTM)                (None, 60)                21840
_________________________________________________________________
dense (Dense)                (None, 5)                 305
=================================================================
Total params: 97,985
Trainable params: 97,985
Non-trainable params: 0
_________________________________________________________________
```

**Figure 4.19 LSTM design**

### 4.10.2 Test training

Initially, the model is run over 1,000 epochs. In order to prevent overfitting, an early stopping checkpoint is set up, with a patience of 10 epochs. This will monitor the validation loss until it has reached the minimum error over 10 epochs.
The input shape into this model is (764, 1000, 59).
Adjust the patience of early stopping to examine its effect on the results.

- Print the results of the training predicted and actuals
- Print future results.
- These are almost always negative.

The activation function for all experiments is the tanh function, which is the hyperbolic tangent function.

As predicting one day out is the most accurate, several different layer structures are used on this one day out prediction.

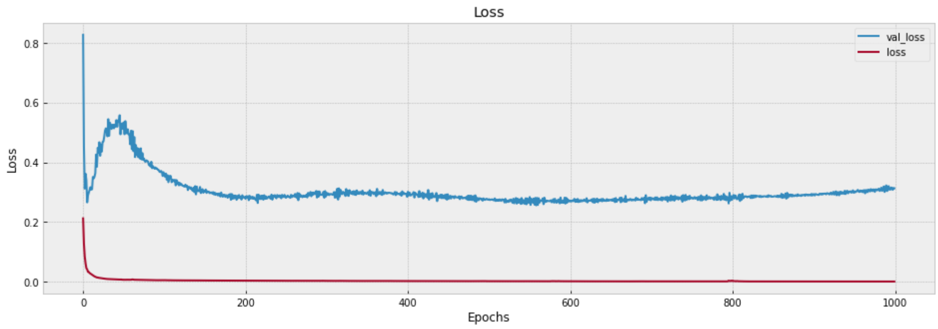


**Figure 4.20 LSTM training loss over 1000 epochs without early stopping**

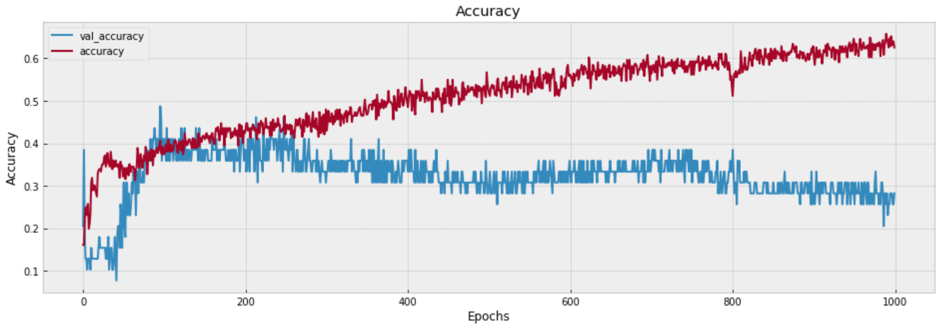


**Figure 4.21 LSTM training accuracy over 1000 epochs**

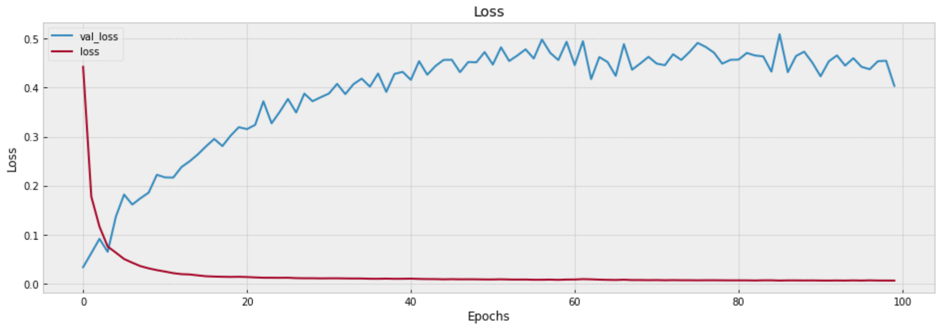


**Figure 4.22 LSTM loss with early stopping**

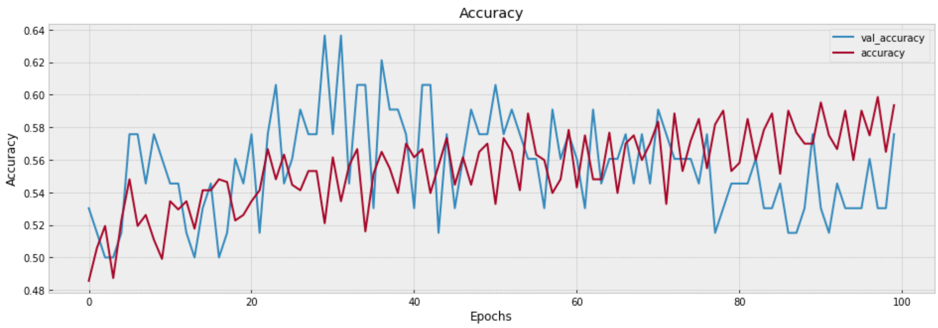


**Figure 4.23 LSTM Accuracy with early stopping**


**Figure 4.24 LSTM 5-day out prediction of BTC**

### 4.10.3Results

| | Day | Layers | Learn back | Features | | Optim | batch size | val % | MSE | Accuracy |
|---|---|---|---|---|---|---|---|---|---|---|
| BTC | 1 | 4- layer | 90 | 59 | tanh | adam | 128 | 0.1 | 0.03008216 | 0.6130501 |
| | 1 | 5- layer | 90 | 59 | tanh | adam | 128 | 0.1 | 0.07320374 | 0.59028834 |
| | 1 | 5- layer | 1000 | 59 | tanh | adam | 128 | 0.1 | 0.00788131 | 0.5431937 |
| | 5 | 5- layer | 365 | 59 | tanh | adam | 128 | 0.1 | 0.03272717 | 0.58792651 |
| | 5 | 5- layer | 500 | 59 | tanh | adam | 128 | 0.1 | 0.02589425 | 0.57317072 |
| | 5 | 4-layer | 90 | 59 | tanh | adam | 128 | 0.1 | 0.03598068 | 0.39786586 |
| | 5 | 4- layer | 90 | 59 | tanh | adam | 128 | 0.1 | 0.06490959 | 0.18858954 |
| | 5 | 4- layer | 500 | 59 | tanh | adam | 128 | 0.1 | 0.00526618 | 0.81522602 |
| | 5 | 4- layer | 1000 | 59 | tanh | adam | 128 | 0.1 | 0.00910519 | 0.69382393 |
| | 30 | 3-layer | 90 | 59 | tanh | adam | 128 | 0.1 | 0.02180129 | 0.21236134 |
| | 30 | 4- layer | 90 | 59 | tanh | adam | 128 | 0.1 | 0.04426183 | 0.21711569 |
| | 30 | 5- layer | 90 | 59 | tanh | adam | 128 | 0.1 | 0.04168136 | 0.37083992 |
| | 30 | 5-layer | 1000 | 59 | tanh | adam | 128 | 0.1 | 0.01068712 | 0.14945652 |

**Table 4.10 BTC results of LSTM**

| | Day | Layers | Learn back | Features | Activ | Optim | batch size | val % | MSE | Accuracy |
|---|---|---|---|---|---|---|---|---|---|---|
| ETH | 1 | 4- layer | 750 | 61 | tanh | adam | 128 | 0.1 | 0.00710782 | 0.57297832 |
| | 1 | 4- layer | 120 | 61 | tanh | adam | 128 | 0.1 | 0.01145179 | 0.54257905 |
| | 1 | 5- layer | 750 | 61 | tanh | adam | 128 | 0.1 | 0.02211106 | 0.50788957 |
| | 1 | 5- layer; dropo | 750 | 61 | tanh | adam | 128 | 0.1 | 0.00920377 | 0.56213015 |
| | 1 | 10- layer; drop | 750 | 61 | tanh | adam | 128 | 0.1 | 0.0902662 | 0.53155816 |
| | 1 | 4- layer | 1500 | 61 | tanh | adam | 128 | 0.1 | 0.02944624 | 0.594697 |
| | 5 | 4- layer | 1500 | 61 | tanh | adam | 128 | 0.1 | 0.03101055 | 0.32950193 |
| | 5 | 4- layer | 750 | 61 | tanh | adam | 128 | 0.1 | 0.01062562 | 0.26013848 |
| | 5 | 4- layer | 120 | 61 | tanh | adam | 128 | 0.1 | 0.01079021 | 0.27361366 |
| | 30 | 4- layer | 1500 | 61 | tanh | adam | 128 | 0.1 | 0.00996461 | 0.23728813 |
| | 30 | 4- layer | 750 | 61 | tanh | adam | 128 | 0.1 | 0.05582715 | 0.11561866 |
| | 30 | 4- layer | 120 | 61 | tanh | adam | 128 | 0.1 | 0.01621604 | 0.13490099 |

**Table 4.11 ETH results of LSTM**

### 4.10.4 Prediction

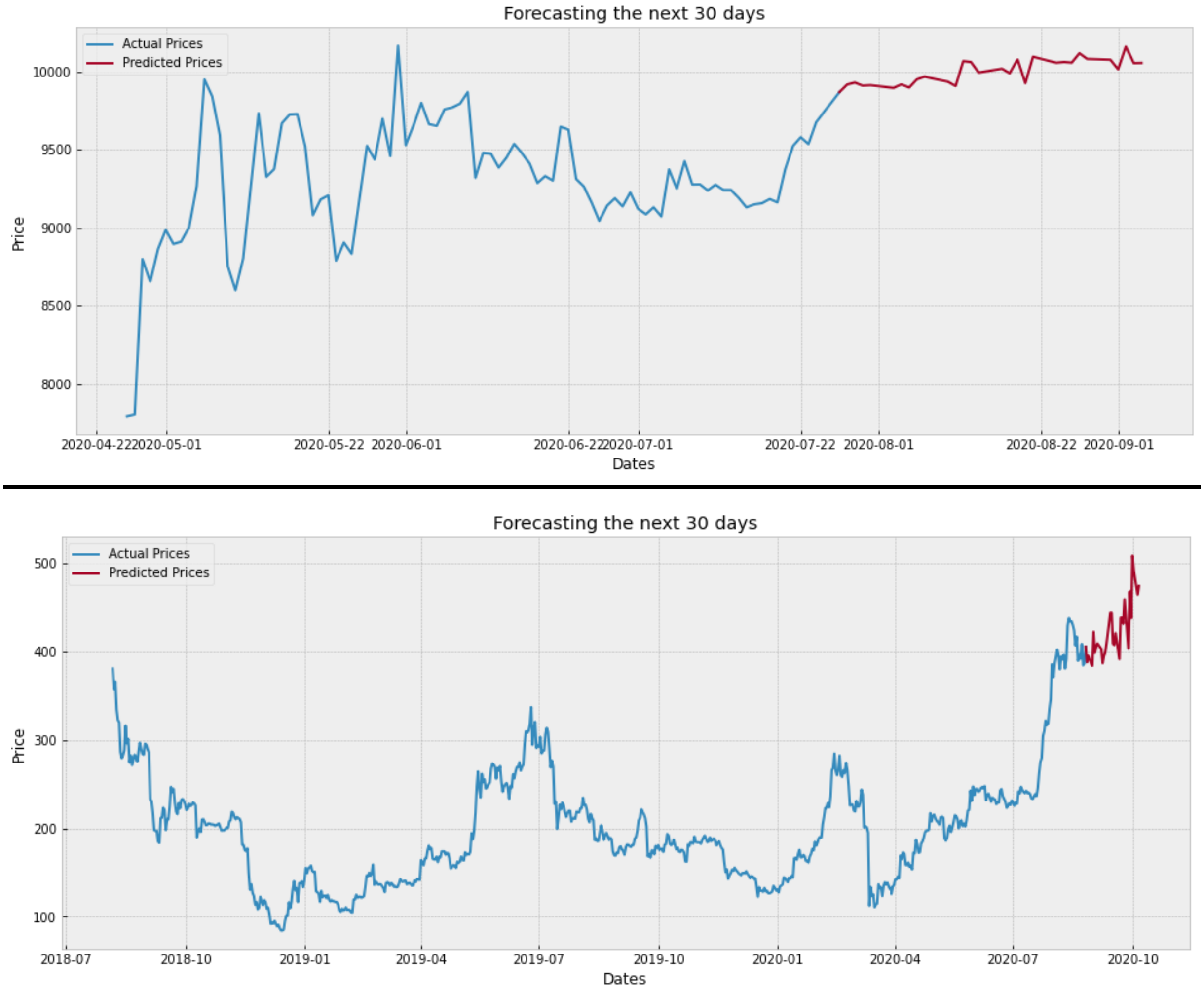


**Figure 4.25 BTC & ETH Forecasting next 30-days: 5-layers; 750 training points.**

### 4.10.5 Evaluation

The effect of the number of layers, although may not have a huge impact on the metrics such as MSE and Accuracy, has a huge effect on the next close prediction when there is no data to compare it to.

It can be seen in Figure 4.26 Predicting , that the next two days prediction is skyrocketing to over double the current value, this is from the model with 12-layers, a drop out after every two LSTM layers, and looking back 750 days to train out 2 days. The results of this model was, accuracy of 53.16% and an MSE of 0.09027, comparatively, a basic model with no dropout and 5-layers has an accuracy of 51.58% and an MSE of 0.01796, with the Figure 4.27 Predicting 1-day out 5-layers, predicting a negative value for the next two days.


**Figure 4.26 Predicting 1-day out– 12 layers**


**Figure 4.27 Predicting 1-day out 5-layers**

This is an example which can be seen across all of the models which were run, there is no stability with predicted direction or value, across the same data with a slightly different configuration, of layers and learning days.

Therefore, although statistically there are some figures within Table 4.10 BTC results of LSTM and Table 4.11 ETH results of LSTM, with above 50% prediction accuracy, when they are tested forward, they are not accurate.

# 5. EVALUATION

In order to evaluate the models comparatively, graphs will be presented, using the accuracy value for each of the experiments run and then the score and MSE for the recurrent neural networks.

## 5.1 Bitcoin results

### 5.1.1 BTC 1 out prediction results

From Figure 5.1 BTC 1-day prediction results scores, that the models do relatively well, meaning that all but one are above 50%, so more likely to be correct than a guess.

The most interesting aspect of these results is the highest accuracy predictions come from the linear and Bayesian regression.

This is likely due to the high Hurst Exponent, for Bitcoin, the Hurst exponent for the entire dataset is 0.9235, therefore, it is almost completely trending, so it is unsurprising that this is the best prediction for Bitcoin one day out, within graphs it can also be seen that trends occur in Bitcoin very often and although the magnitude of the trend is difficult to predict, there are clear indications of it being linear.

The neural network model within Machine learning which is using ReLU as its activation function and a formation of (8,8) has a highly negative prediction accuracy.

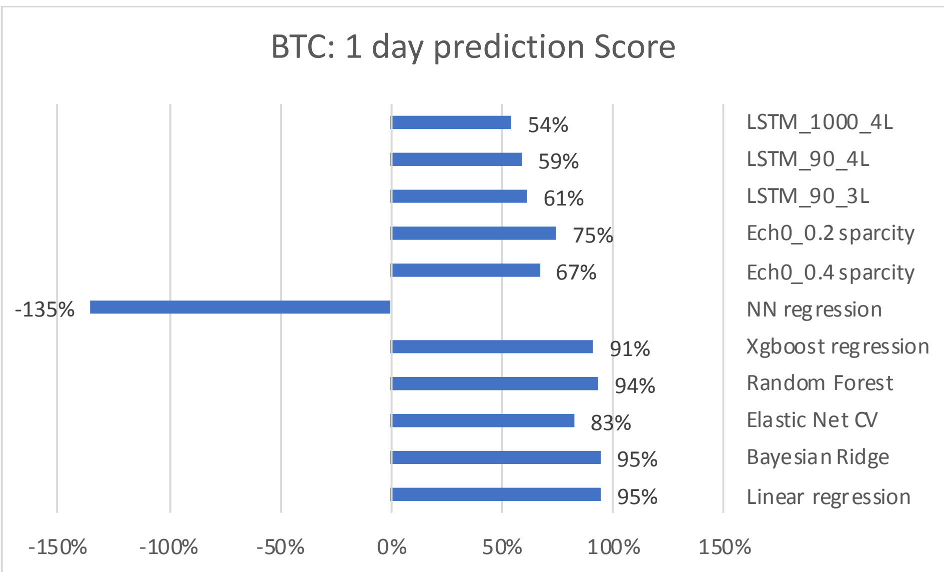


**Figure 5.1 BTC 1-day prediction**

In order to understand the recurrent neural networks in more detail, Figure 5.2 BTC 1-day prediction RNN is produced, here we can see that although the MSE for LSTM with 1000 learning days and 4-layers is the lowest, it also has the lowest accuracy score.
This may be due to overtraining.
While the MSE is at its highest for the Echo state networks, it also has the highest accuracy, this is likely due to the chaotic nature of the ESN, with the switching off of nodes within the reservoir, the results are overly dependent on random nodes.

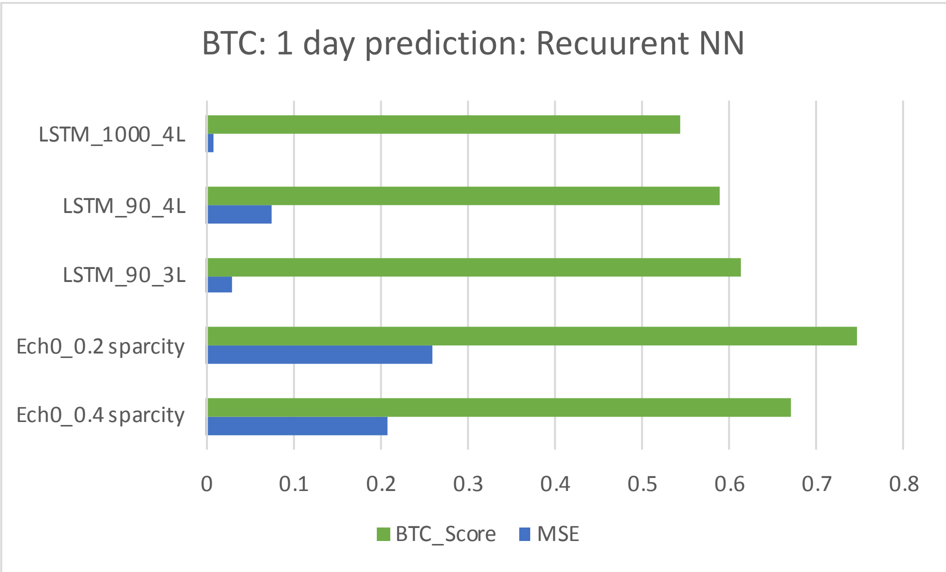


**Figure 5.2 BTC 1-day prediction RNN**

### 5.1.2 BTC 5-day prediction results

The result takes a notable drop in accuracy score from day one predictions.
We can also see that the machine learning models are still doing statistically well, while the neural network regression goes from -135% to -95%, it is clear that the ReLU activation and the Neural network without any backtesting is not a viable option for predictions.

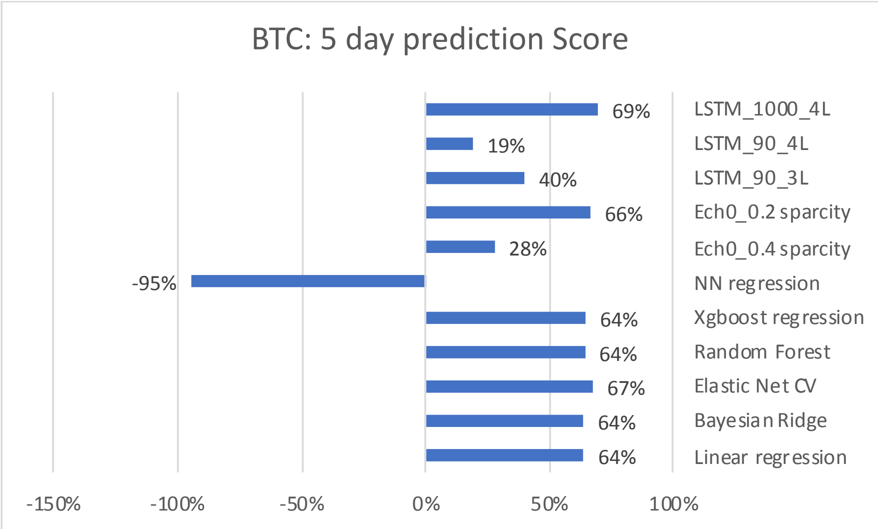


**Figure 5.3 BTC 5-day out predictions**

Interestingly here the length of training of the LSTM has a positive effect on the prediction score, with a sharp difference between the 4-layer trained off 90 days and the 4-layered trained off 1000 days. We can also see that the accuracy of the ESN with sparsity 0.2 also outperforms the 0.4 sparsity accuracy in the 5-day prediction, as it did in the one-day prediction

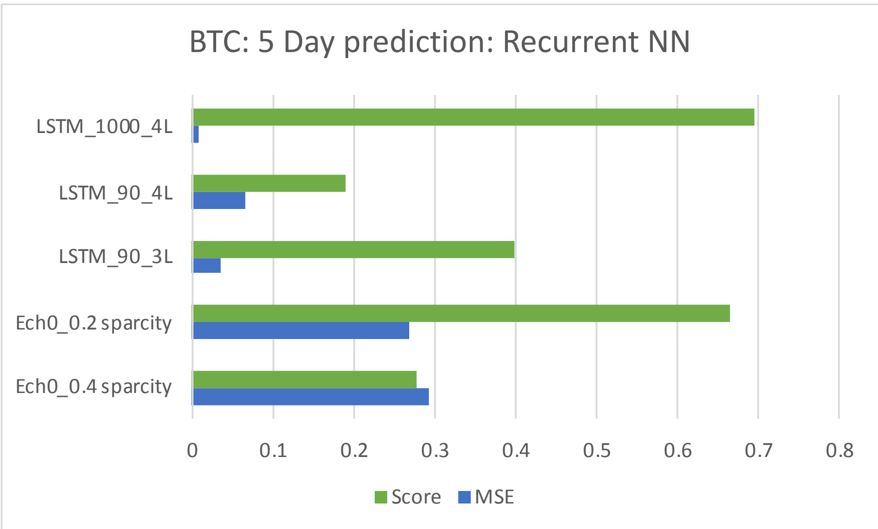


**Figure 5.4 BTC 5-day RNN results**

### 5.1.3 BTC 30-day prediction results

As expected the ability to predict 30 days into the future for Bitcoin is incredibly difficult.

Although the regression neural network has been underperforming on every model, it is surprising that the Echo state network at 0.2 sparsity is performing so poorly for this prediction.

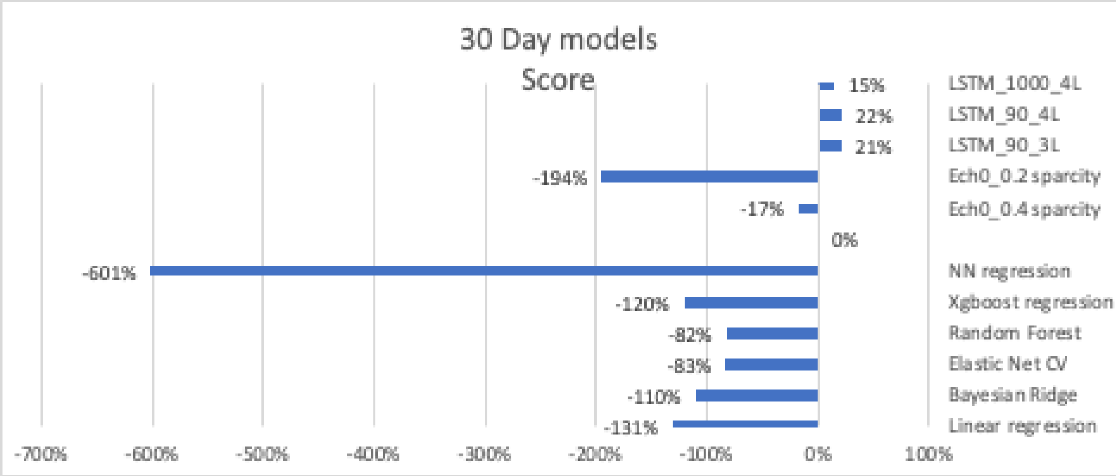


**Figure 5.5 BTC 30-day out predictions**

The MSE although high for both Echo state networks shows the worst results out of the recurrent neural networks.

Also surprising, is the low accuracy for the LSTM learning from 1000 days, a longer dataset with more stability is needed for better results with this length of prediction.

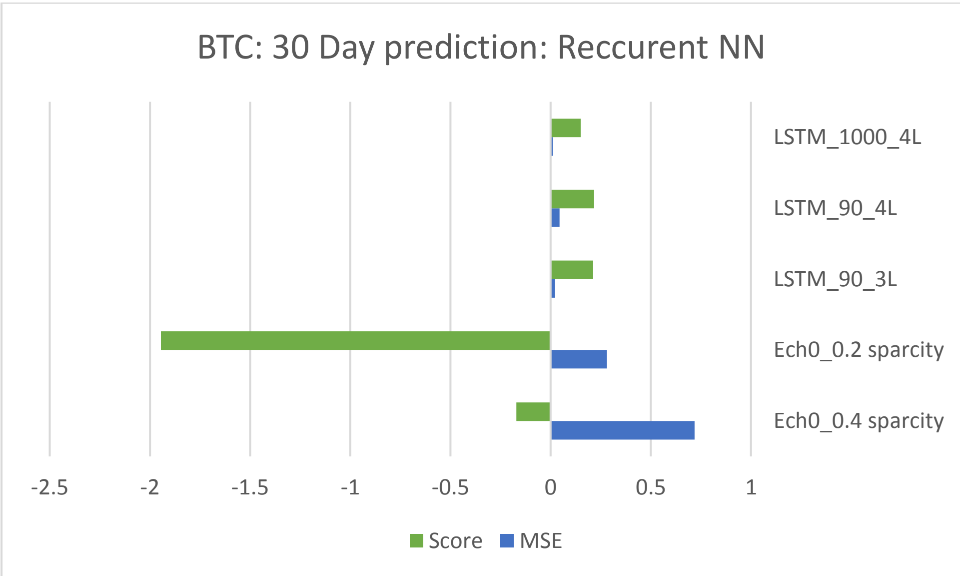


**Figure 5.6 BTC 30-day RNN results**

## 5.2 ETH results

From Figure 5.7 ETH 1-day prediction results scores, that the models do relatively well, meaning that all but one are above 50%, so more likely to be correct than a guess.

The most interesting aspect of these results is the highest accuracy predictions come from the Linear and Bayesian regression and XgBoost

This is likely due to the high Hurst Exponent, for ETH the Hurst exponent for the entire dataset is 0.909,9 therefore, it is almost completely trending, unsurprisingly, hat this is the best prediction for ETH one day out.

The neural network model within Machine learning which is using ReLU as its activation function and a formation of (8,8) has a positive result for this dataset, outperforming the LSTM models significantly.

### 5.2.1 ETH 1-day prediction results

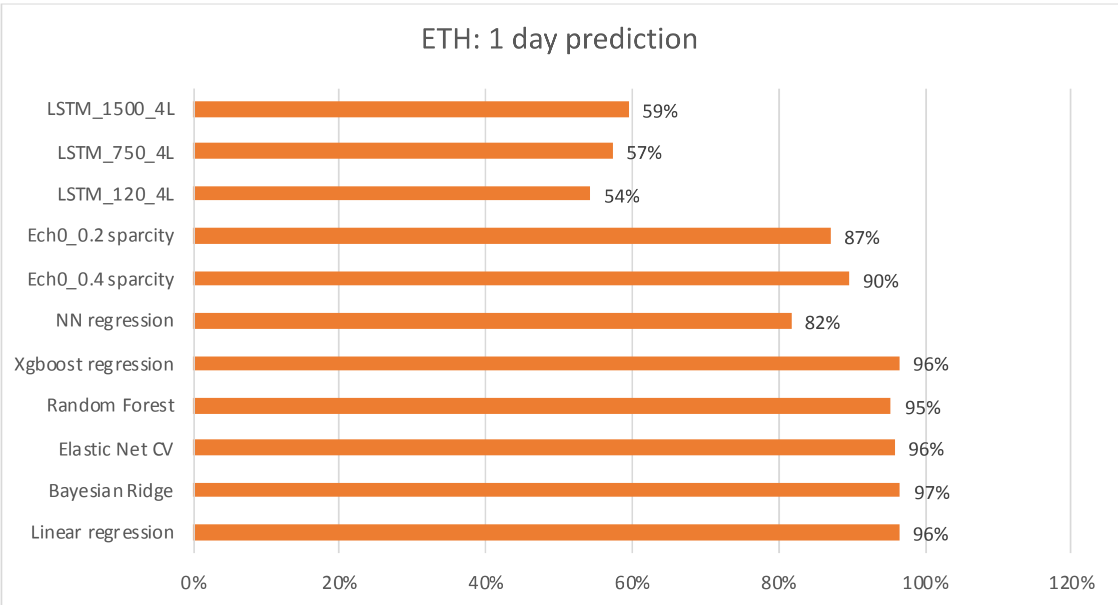


**Figure 5.7 ETH 1-day prediction**

The Echo State Network proves to be a good tool for predicting the ETH closing price of the next day, the model performs at 90% for sparsity, although the concern is that the linear model, which is the output for the Echo state network, produces better results than the ESN.

The LSTM learning from 120 days, products the poorest result, but also has a low MSE, this is likely due to overtraining and the models should be adjusted with dropout. The models may also be negatively affected by early stopping. Within future work, there should be an increased drop out and increased patience for early stopping.

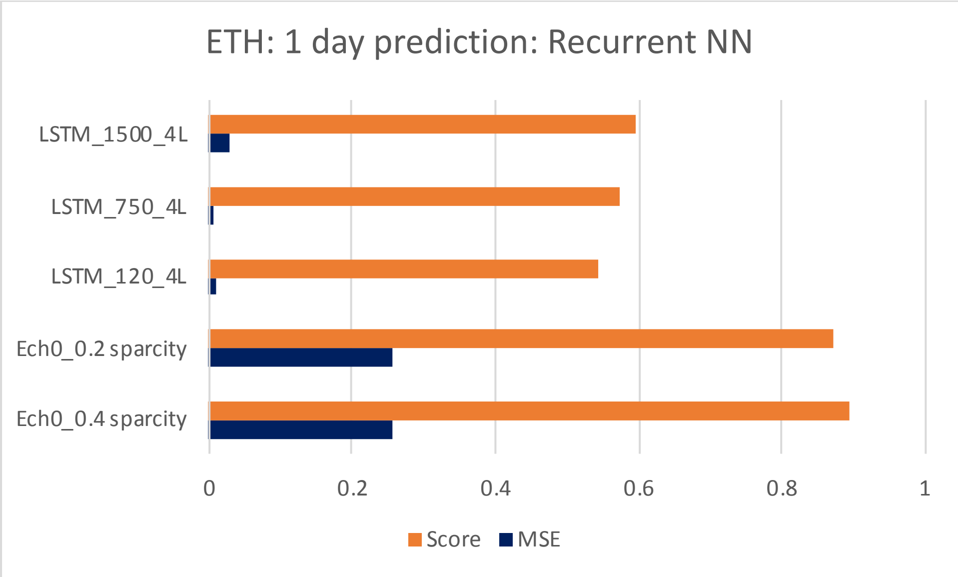


**Figure 5.8 ETH 1-day RNN results**

### 5.2.2 ETH 5 day out prediction

ElasticNetCV CV and XgBoost have the best performance within this 5-day out prediction.

With the lowest scores on performance from LSTM 750 day training set with 4-layers.

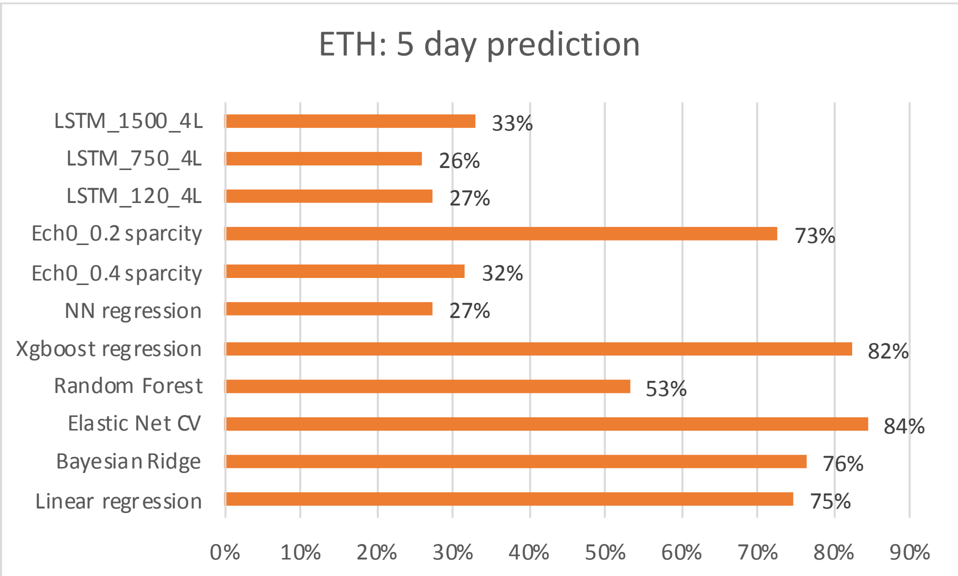


**Figure 5.9 ETH 5-day out prediction results**

Of the recurrent neural networks, Echo state network with 0.2 sparsity is the best performing, although the MSE is high, it can be seen from the graphs that on initialization the ESN does poorly but often steadies itself out.
Figure 4.16 ETH predictions Learning days 100 particularly highlight this, with the data zoomed in on the 100 days.

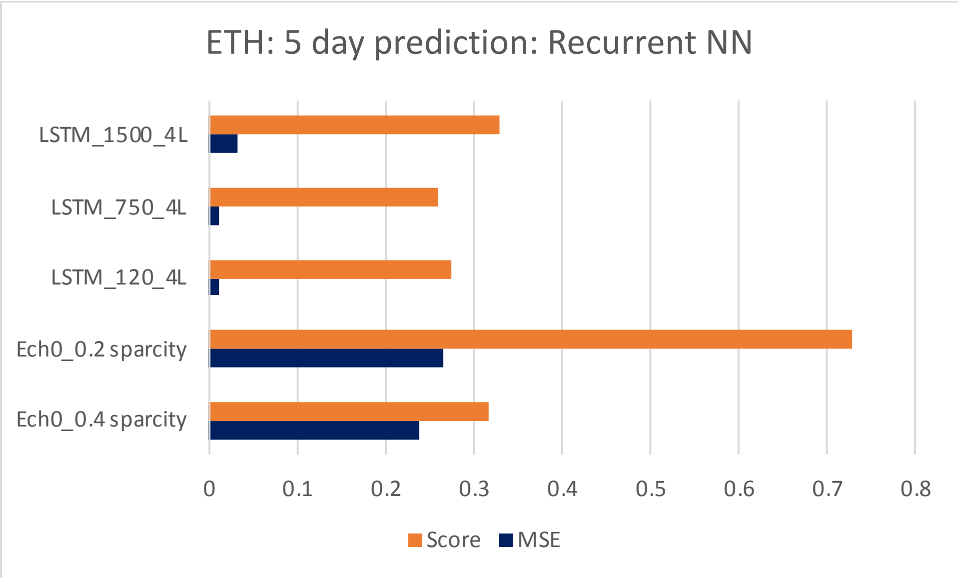


**Figure 5.10 ETH 5-day out results**

### 5.2.3 ETH 30 day out predictions.

As expected the 30 day out prediction performs poorly, with the linear and Bayesian regression being the most negative compared to when BTC was being predicted. Surprisingly the LSTM with the lowest number of training days to learn from has the highest accuracy and lowest MSE.

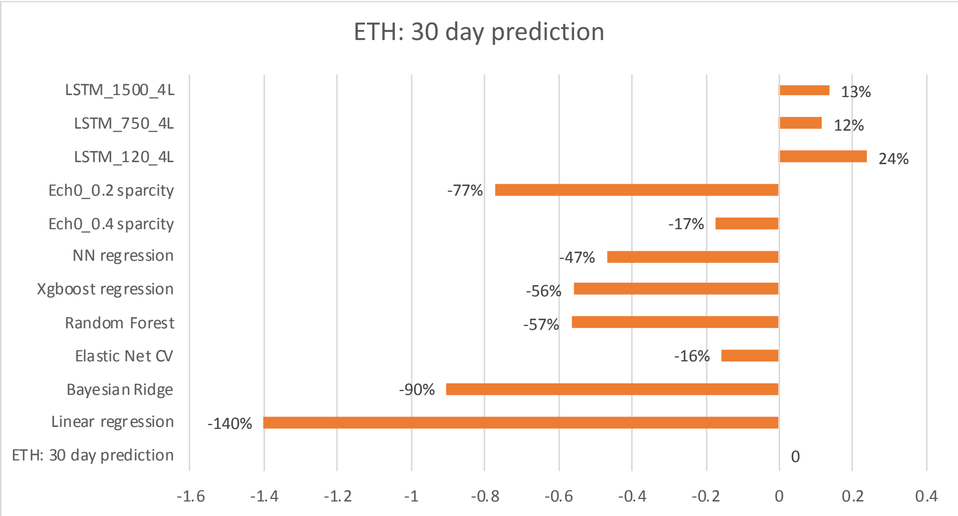


**Figure 5.11 ETH 30-day out results**

The echo state networks underperform on this prediction set, which is disappointing, but shows future work into using tuning the reservoir nodes may be needed within this model.

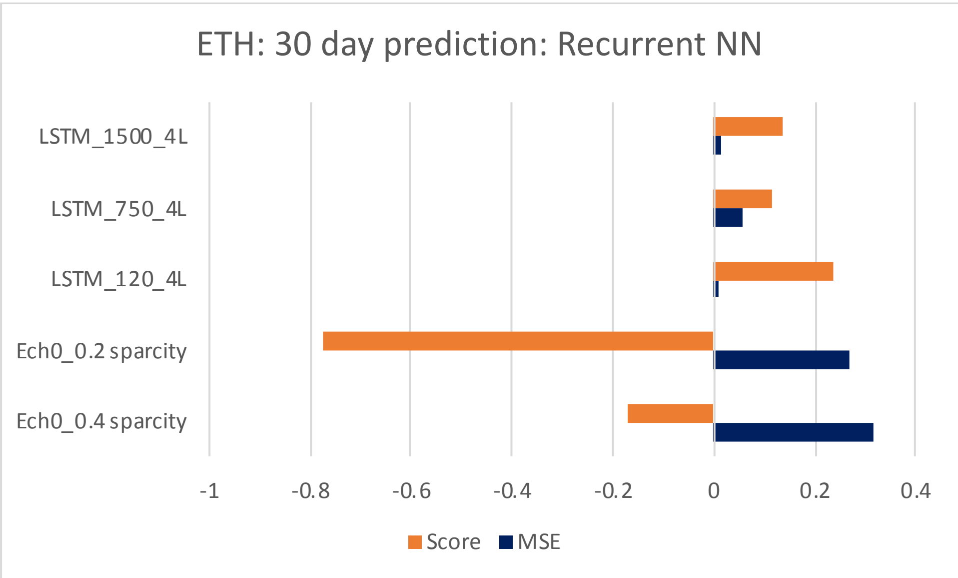


**Figure 5.12 ETH 30-day out RNN results**

# 6. CONCLUSION

This chapter will provide an overview of the study. The research aim, question and the insights gained from the process of answering these questions.
This chapter will present the research overview/problem, design and implementation and give context to the evaluation and results, to provide recommendations on future work.

## 6.1 Research Overview

As the cryptocurrency market continues to fluctuation as wildly as it has this year, it will remain difficult to predict. Due to the events of this year within the global economy with a global pandemic, predictive models based on technical analysis have shown their flaws. This research aimed to predict models for next day, next 5 days and the month, with the goal to provide a view of the best predictive model for your needs.

This study provided a comprehensive study on the use of technical indicator feature selection, using exploratory and permutation importance to pick specific features for the individual currencies
For predictions machine learning pipelines and two types of recurrent neural networks, with the LSTM having the potential to get stuck in the vanishing gradient, the ESN provides a chaotic neural network which can be used to ensure there isn't an issue with overfitting.

With the vast set of literature presented in academia on the value of cryptocurrencies and on forecasting techniques that can be used, it was noted from all the literature that due to the chaotic nature of cryptocurrency markets, there is a significant fear of overfitting the model, therefore a neural network with a chaotic element may provide accurate predictions without the fear of overfitting. It was also noted from reading these research papers that there are very few research papers on Ether, this is likely due to how correlated it is with Bitcoin and while Bitcoin has more data it can train models more, that is why for this experiment, the data range for training the models was the same, to ensure that like with like could be compared.

Based on the literature review finding the overall goal of the study was to examine the predictability *Bitcoin and Ethereum short-, medium- and long-term direction of pricing, can be predicted by only using technical analysis with Machine learning and Neural Network.* By examining 2 cryptocurrencies and 80 technical indicators to assist in forecasting, from strength indicators, oscillators, momentum indicators and mean reversion indicators, better insights into the market were developed and fed into models.

This research project aimed to highlight the importance and evaluate the usefulness of different models in chaotic time series prediction.

## 6.2 Design/Experimentation, Evaluation & Results

The experiment was designed so that the data input could be altered to any cryptocurrency or stock, this was a key design choice in allowing the model to pick from so many technical indicators, and then running feature selection.

Each model takes in only the parameters given, with little manipulation to the code required. With hyperparameters for the Echo State network by running a range of the values and selecting those with the minimum MSE.

### 6.2.1 Overview of design

Data pre-processing:

- Key packages: NumPy, pandas, matplotlib

Data processing

- Key packages: TA-lib, eli5,

Model constructions:

- Echo State Network
  - Key packages: pyESN
- LSTM
  - Key packages: TensorFlow
- Pipelines created with sklearn
  - Linear Regression
  - Bayesian Ridge
  - ElasticNetCV
  - Random Forest Regressor – using n_estimator of 500
  - XgBoost Regression – extreme gradient boosting
  - Neural Network
    - Activation function – ‘ReLU’
    - Optimizer – ‘adam’

Model evaluation – visual

Once the models were implemented an evaluation of the individual models was completed and followed by an evaluation of all of the models over their prediction target time.

The evaluation of this research highlighted the need for strategies to not only rely on one type of model but for the continuous work on the model and the importance of feature selection specific to the models.

## 6.3 Contributions and impact

"Markets do not follow a random walk and are persistent, which is inconsistent with market efficiency"(Caporale et al., 2018), this makes predictive models easier, as the markets are not dependent on new variables to dictate their price, the influence of external factors are reduced. As was proven through the Linear regression model scores within the 1-day and 5-day prediction of both models, the impact of the Hurst Exponent being ($H>0.5$) shows that it is easier to predict the cryptocurrency market in short term burst now, that if it were to get to a random walk stage.

This study provides insights into the use of Echo State Networks for predicting cryptocurrencies, which is not a deeply explored area of research from my findings, although Echo state networks have been used in the stock market and forex predictions.

## 6.4 Future Work & recommendations

There is an immense amount of future work that can come from this study. This is a rapidly changing area of finance.

Future work within this area could include

- The LSTM learning from 120 days, products the poorest result, but also has a low MSE, this is likely due to overtraining and the models should be adjusted with dropout. The models may also be negatively affected by early stopping. Within future work, there should be an increased drop out and increased patience for early stopping.
- Echo State network future work, a study on adapting the number of reservoir nodes with the sparsity
- Implementing a portfolio based on the predictions from the models to analyse which model predictions over a set period and trading rules such as (if ESN prediction >5%, buy, ELSE (if ESN prediction <5%, sell), ELSE, hold)

# 7. BIBLIOGRAPHY


Adachi, M., & Aihara, K. (1997). Associative Dynamics in a Chaotic Neural Network. *Neural Networks*, *10*(1), 83–98. https://doi.org/10.1016/S0893-6080(96)00061-5

Alessandretti, L., ElBahrawy, A., Aiello, L. M., & Baronchelli, A. (2018, November 4). *Anticipating Cryptocurrency Prices Using Machine Learning* [Research Article]. Complexity; Hindawi. https://doi.org/10.1155/2018/8983590

Amirat, Dr. A., & Zaidi, M. Dr. (2016). Forecasting Stock Market Trends by Logistic Regression and Neural Networks: Evidence from KSA Stock Market. *Euro-Asian Journal of Economics and Finance*, *Volume 4*(Issue 2), 50–58.

Böhme, R., Christin, N., Edelman, B., & Moore, T. (2015). Bitcoin: Economics, Technology, and Governance. *Journal of Economic Perspectives*, *29*(2), 213–238. https://doi.org/10.1257/jep.29.2.213

Caporale, G. M., Gil-Alana, L., & Plastun, A. (2018). Persistence in the cryptocurrency market. *Research in International Business and Finance*, *46*, 141–148. https://doi.org/10.1016/j.ribaf.2018.01.002

Carbone, A., Castelli, G., & Stanley, H. E. (2004). Time-dependent Hurst exponent in financial time series. *Physica A: Statistical Mechanics and Its Applications*, *344*(1), 267–271. https://doi.org/10.1016/j.physa.2004.06.130

Chatzis, S. P., Siakoulis, V., Petropoulos, A., Stavroulakis, E., & Vlachogiannakis, N. (2018). Forecasting stock market crisis events using deep and statistical machine learning techniques. *Expert Systems with Applications*, *112*, 353–371. https://doi.org/10.1016/j.eswa.2018.06.032

Cheah, E.-T., & Fry, J. (2015). Speculative bubbles in Bitcoin markets? An empirical investigation into the fundamental value of Bitcoin. *Economics Letters*, *130*, 32–36. https://doi.org/10.1016/j.econlet.2015.02.029

Cosme Andrieu, E., & Steeb, W.-H. (2005, January 4). *Lyapunov Exponents, Hyperchaos and Hurst Exponent*. https://www.researchgate.net/publication/267841300_Ljapunov_Exponents_Hyperchaos_and_Hurst_Exponent

Dai, W., Wu, J.-Y., & Lu, C.-J. (2012). Combining nonlinear independent component analysis and neural network for the prediction of Asian stock market indexes. *Expert Systems with Applications*, *39*(4), 4444–4452. https://doi.org/10.1016/j.eswa.2011.09.145

Dutta, A., Kumar, S., & Basu, M. (2020a). A Gated Recurrent Unit Approach to Bitcoin Price Prediction. *Journal of Risk and Financial Management*, *13*(2), 23. https://doi.org/10.3390/jrfm13020023

Dutta, A., Kumar, S., & Basu, M. (2020b). A Gated Recurrent Unit Approach to Bitcoin Price Prediction. *Journal of Risk and Financial Management*, *13*(2), 23. https://doi.org/10.3390/jrfm13020023

Enke, D., & Thawornwong, S. (2005). The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications*, *29*(4), 927–940. https://doi.org/10.1016/j.eswa.2005.06.024

Gradwell, P. (2020, March 18). *What Is Going On in the Bitcoin Market?* Chainalysis. https://blog.chainalysis.com/reports/bitcoin-market-march-2020

Greaves, A., & Au, B. (2015). *Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin*. 8.

Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert Systems with Applications*, *38*(8), 10389–10397. https://doi.org/10.1016/j.eswa.2011.02.068

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Huang, J.-Z., Huang, W., & Ni, J. (2019). Predicting bitcoin returns using high-dimensional technical indicators. *The Journal of Finance and Data Science*, *5*(3), 140–155. https://doi.org/10.1016/j.jfds.2018.10.001

Hurst, H. E. (1951). Long-term storage of reservoirs: An experimental study. *Transactions of the American Society of Civil Engineers*, *116*, 770–799.

Indera, N. I., Yassin, I. M., Zabidi, A., & Rizman, Z. I. (2017). Non-linear Autoregressive with Exogeneous input (narx) bitcoin price prediction model using PSO-optimized parameters and moving average technical indicators. *Journal of Fundamental and Applied Sciences*, *9*(3S), 791–808. https://doi.org/10.4314/jfas.v9i3S.61

Jaeger, H. (2004). Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science*, *304*(5667), 78–80. https://doi.org/10.1126/science.1091277

Jaeger, H., & Haas, H. (2004). Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science*. https://doi.org/10.1126/science.1091277

Jain, A., & Kumar, A. M. (2007). Hybrid neural network models for hydrologic time series forecasting. *Applied Soft Computing*, *7*(2), 585–592. https://doi.org/10.1016/j.asoc.2006.03.002

Jang, H., & Lee, J. (2018). An Empirical Study on Modeling and Prediction of Bitcoin Prices With Bayesian Neural Networks Based on Blockchain Information. *IEEE Access*, *6*, 5427–5437. https://doi.org/10.1109/ACCESS.2017.2779181

Korndörfer, C. (2015). *PyESN*.

Kurihara, Y., & Fukushima, A. (2017). *The Market Efficiency of Bitcoin: A Weekly Anomaly Perspective*. 8.

Lahmiri, S., & Bekiros, S. (2019). Cryptocurrency forecasting with deep learning chaotic neural networks. *Chaos, Solitons & Fractals*, *118*, 35–40. https://doi.org/10.1016/j.chaos.2018.11.014

Lahmiri, S., & Bekiros, S. (2020). Intelligent forecasting with machine learning trading systems in chaotic intraday Bitcoin market. *Chaos, Solitons & Fractals*, *133*, 109641. https://doi.org/10.1016/j.chaos.2020.109641

Lee, R. S. T. (2019). COSMOS trader – Chaotic Neuro-oscillatory multiagent financial prediction and trading system. *The Journal of Finance and Data Science*, *5*(2), 61–82. https://doi.org/10.1016/j.jfds.2019.01.001

Lendasse, A., Bodt, E. de, Wertz, V., & Verleysen, M. (2000). Non-linear financial time series forecasting—Application to the Bel 20 stock market index. *European Journal of Economic and Social Systems*, *14*(1), 81–91. https://doi.org/10.1051/ejess:2000110

Li, D., Han, M., & Wang, J. (2012). Chaotic Time Series Prediction Based on a Novel Robust Echo State Network. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(5), 787–799. https://doi.org/10.1109/TNNLS.2012.2188414

Li, Q., & Lin, R.-C. (2016, December 6). *A New Approach for Chaotic Time Series Prediction Using Recurrent Neural Network* [Research Article]. Mathematical Problems in Engineering; Hindawi. https://doi.org/10.1155/2016/3542898

Lin, X., Yang, Z., & Song, Y. (2009). Short-term stock price prediction based on echo state networks. *Expert Systems with Applications*, *36*(3, Part 2), 7313–7317. https://doi.org/10.1016/j.eswa.2008.09.049

Lui, F., Quek, C., & See Ng, G. (2005, July 31). *Neural network model for time series prediction by reinforcement learning*. https://ieeexplore.ieee.org/abstract/document/1555956/

Madan, I., Saluja, S., & Zhao, A. (2015). *Automated Bitcoin Trading via Machine Learning Algorithms*. 5.

Malkiel, B. G. (2003). The Efficient Market Hypothesis and Its Critics. *Journal of Economic Perspectives*, *17*(1), 59–82.

Matta, M., Lunesu, I., & Marchesi, M. (2015). *Bitcoin Spread Prediction Using Social And Web Search Media*. 10.

Nahil, A., & Lyhyaoui, A. (2018). Short-term stock price forecasting using kernel principal component analysis and support vector machines: The case of Casablanca stock exchange. *Procedia Computer Science*, *127*, 161–169. https://doi.org/10.1016/j.procs.2018.01.111

Ning, B., Wu, J., Peng, H., & Zhao, J. (2009). Using Chaotic Neural Network to Forecast Stock Index. In W. Yu, H. He, & N. Zhang (Eds.), *Advances in Neural Networks – ISNN 2009* (Vol. 5551, pp. 870–876). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-01507-6_98

Qiu, M., & Song, Y. (2016). Predicting the Direction of Stock Market Index Movement Using an Optimized Artificial Neural Network Model. *PLOS ONE*, *11*(5), e0155133. https://doi.org/10.1371/journal.pone.0155133

Qiu, Y., & Lee, R. S. T. (2019). A Hybrid Chaotic Oscillatory Neural Network (HCONN) Based Financial Time Series Prediction System. *IOP Conference Series: Materials Science and Engineering*, *646*(1), 012024. https://doi.org/10.1088/1757-899X/646/1/012024

Raiborn, C. A., & Sivitanides, M. (2015). Accounting issues related to bitcoins. *The Journal of Corporate Accounting & Finance*, *26*(2).

Rauchs, M., Blandin, A., Klein, K., Pieters, G., Recanatini, M., & Zhang, B. (2018). 2nd Global Cryptoasset Benchmarking Study. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3306125

Rosenstein, M. T., Collins, J. J., & De Luca, C. J. (1993). A practical method for calculating largest Lyapunov exponents from small data sets. *Physica D: Nonlinear Phenomena*, *65*(1), 117–134. https://doi.org/10.1016/0167-2789(93)90009-P

Sang, C., & Di Pierro, M. (2019). Improving trading technical analysis with TensorFlow Long Short-Term Memory (LSTM) Neural Network. *The Journal of Finance and Data Science*, *5*(1), 1–11. https://doi.org/10.1016/j.jfds.2018.10.003

Sin, E., & Wang, L. (2017). Bitcoin price prediction using ensembles of neural networks. *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 666–671. https://doi.org/10.1109/FSKD.2017.8393351

Skowronski, M. D., & Harris, J. G. (2006). Minimum mean squared error time series classification using an echo state network prediction model. *2006 IEEE*

*International Symposium on Circuits and Systems*, 4 pp. – 3156. https://doi.org/10.1109/ISCAS.2006.1693294

Tran, V. L., & Leirvik, T. (2020). Efficiency in the markets of crypto-currencies. *Finance Research Letters*, *35*, 101382. https://doi.org/10.1016/j.frl.2019.101382

Zainuddin, N. H., Lola, M. S., Djauhari, M. A., Yusof, F., Ramlee, M. N. A., Deraman, A., Ibrahim, Y., & Abdullah, M. T. (2019). Improvement of time forecasting models using a novel hybridization of bootstrap and double bootstrap artificial neural networks. *Applied Soft Computing*, *84*, 105676. https://doi.org/10.1016/j.asoc.2019.105676

Zerroug, A., Terrissa, L., Faure, A., & Greah, L. (2013). *Chaotic Dynamical Behavior of Recurrent Neural Network*. 12.